1996

Quantisation mechanisms in multi-protoype waveform coding

Duong Hong Pham
University of Wollongong

QUANTISATION MECHANISMS IN MULTI-PROTOTYPE WAVEFORM CODING

A thesis submitted in partial fulfilment of the
requirements for the award of the degree of

MASTER OF ENGINEERING IN
TELECOMMUNICATIONS ENGINEERING
(Honours)

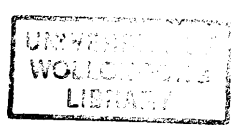
from the

UNIVERSITY OF WOLLONGONG

by

DUONG HONG PHAM

Bachelor of Science (Physics)
University of Hanoi, Vietnam (1990)



DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

July 1996

Declaration

This is to certify that the work reported in this thesis was done by the author, unless specified otherwise, and that no part of it has been submitted in a thesis to any other university or similar institution.

Duong Hong Pham

ABSTRACT

Prototype Waveform Coding is one of the most promising methods for speech coding at low bit rates over telecommunications networks. This thesis investigates quantisation mechanisms in Multi-Prototype Waveform (MPW) coding, and two prototype waveform quantisation algorithms for speech coding at bit rates of 2.4kb/s are proposed. Speech coders based on these algorithms have been found to be capable of producing coded speech with equivalent perceptual quality to that generated by the US 1016 Federal Standard CELP-4.8kb/s algorithm.

The two proposed prototype waveform quantisation algorithms are based on Prototype Waveform Interpolation (PWI). The first algorithm is in an open loop architecture (Open Loop Quantisation). In this algorithm, the speech residual is represented as a series of prototype waveforms (PWs). The PWs are extracted in both voiced and unvoiced speech, time aligned and quantised and, at the receiver, the excitation is reconstructed by smooth interpolation between them. For low bit rate coding, the PW is decomposed into a slowly evolving waveform (SEW) and a rapidly evolving waveform (REW). The SEW is coded using vector quantisation on both magnitude and phase spectra. The SEW codebook search is based on the best matching of the SEW and the SEW codebook vector. The REW phase spectra is not quantised, but it is recovered using Gaussian noise. The REW magnitude spectra, on the other hand, can be either quantised with a certain update rate or only derived according to SEW behaviours.

The second prototype waveform quantisation algorithm is designed in an analysis-by-synthesis architecture (Analysis-by-Synthesis Quantisation). The aim of this algorithm is to improve the Open Loop algorithm. In this technique, the SEW codebook search is based on matching the incoming PW and the candidate PW, which has been constructed from the SEW codebook vector. For the codebook search, the PWs are represented either in the residual domain or in the speech domain, thus a perceptual synthesis filter can be used for speech quality enhancement. For quantisation, rather than decomposing the PW into a SEW and a REW, the Analysis-by-Synthesis Quantisation considers that the PW can be constructed from a SEW and a REW. This quantisation is, therefore, advantageous over the Open Loop Quantisation in terms of perceptual quality, the use of SEW codebooks and other applications.

Both quantisation algorithms were tested along with the US 1016 Federal Standard CELP-4.8kb/s using the Mean Opinion Score measure. The test results show that the speech coded by the new technique is equivalent or better than that generated by the US 1016 Federal Standard CELP-4.8kb/s.

Acknowledgements

I would like to thank my supervisor, Dr Ian Burnett, for his support, guidance and encouragement which have made it possible for me to complete this work. I would also like to thank the Australian Government as this work was supported by a scholarship from AusAID.

I would like to thank Dr Philip Ogunbona for his contribution in our useful discussions on this research. In addition, I would like to acknowledge the assistance of Messrs Mohsen Kahani, Jamshid Shanbehzadeh, Ali Yazdian Varjani, Nguyen Ngoc Phuong, Nguyen Dinh Truong, and other colleagues.

I would like to thank Ms. Maree Fryer for reading the manuscript and members of the staff in the Department of Electrical and Computer Engineering for their assistance throughout my studies.

Finally, I am grateful to my family for their continued and immense support at all times. A special thanks to my close friend, Ms. Nguyen Hong Hanh, for all her support.

Glossary of Acronyms

The following acronyms are used throughout this thesis:

A-by-S	Analysis-by-Synthesis
Av. SNR	Average Signal-to-Noise Ratio
CELP	Code Excited Linear Prediction
DFT	Discrete Fourier Transform
FIR	Finite Impulse Response
IIR	Infinite Impulse Response
LBG	Linde Buzo Gray
LPC	Linear Predictive Coding
LP	Linear Prediction
LSF	Line Spectral Frequency
LTP	Long Term Prediction
MOS	Mean Opinion Score
MPW	Multi-Prototype Waveform
PW	Prototype Waveform
PWI	Prototype Waveform Interpolation
Q. Amp	Quantisation Amplitude
RELP	Residual Excited Linear Prediction
REW	Rapidly Evolving Waveform
SA	Simulated Annealing
SD	Spectral Distortion
Seg. SNR	Segmental Signal-to-Noise Ratio
SEW	Slowly Evolving Waveform
SNR	Signal-to-Noise Ratio

SQ	Scalar Quantisation (Quantiser)
TIMIT	Texas Instruments Massachusetts Institute of Technology Speech Database
VQ	Vector Quantisation (Quantiser)
WI	Waveform Interpolation

Table of Contents

Chapter 1: Introduction

Over the last several years, considerable research has been undertaken in the area of speech coding. In particular, research has focussed on low bit rate speech coding techniques for telecommunications, especially, for digital mobile radio satellite communication systems. The rapid increase in the number of subscribers is providing demand for higher capacity in digital mobile systems. To cater for this, telecommunication groups in America, Europe and Japan have been trying to establish new digital mobile communication standards with greater capacity than the current systems. Due to the limited bandwidth available, the aim, generally, is to halve the current transmission rate of each channel, while still achieving equal or better performance. Recent advancements in Digital Signal Processing (DSP) devices have provided the basis for digital speech coding, both in terms of new algorithm research and practical implementation. Thus, new low bit rate speech coding algorithms, even very complex algorithms, can be efficiently realised and catering for the demands of the new digital mobile systems has become realistic.

1.1 Speech Coding Techniques

Speech coding techniques, historically, have classified speech coders into two types, that is, vocoders and waveform coders [1]. Vocoders are designed using the basic models of speech production and speech perception. In these coders, perceptual parameters of input speech are extracted and then used for reconstruction of the speech. Waveform coders, on the other hand, quantise the speech '*waveform*' and attempt to reproduce the original waveform of input speech. Compared to waveform coders, vocoders are much more dependent on the speech production model, however, they can produce good quality speech at lower bit rates than waveform coders.

Residual Excited Linear Predictive (RELP) coding, Code Excited Linear Predictive (CELP) coding [2,3] and Linear Predictive (LP) coding all belong to the group of vocoding techniques. RELP based coders [4,5] are typically capable of providing high quality speech at bit rates below 16kb/s. Coders operating at these rates are widely used for applications such as digital cellular, aeronautical, maritime and military communications [6]. Currently, CELP coders such as the US Federal Standard 1016 can generate good communication quality speech at rates of 4.8kb/s [7,8], while the speech quality of the 2.4kb/s LPC-10 coder [9] is poor and lacks naturalness. The later is thus used for certain military purposes only.

Frequency domain coders, of which the common coding algorithms are transform coding and sub-band coding, are one class of waveform coder. Transform coders analyse and decompose short segments of input speech into a number of frequency components [10]. The resulting frequency components can be effectively quantised using an adaptive bit allocation scheme. Transform based coders [11], generally, achieve high quality speech at a medium rate of approximately 16kb/s. In sub-band coders, the input speech spectrum is divided into a set of contiguous sub-bands by means of filtering

[10]. A key to achieving high quality in sub-band coding is a dynamic bit allocation scheme for the sub-band outputs. Sub-band coders, like transform coders, can produce good communication quality speech at medium rates such as 8kb/s [12], above 7kb/s [13], and 6.4kb/s [14].

Sinusoidal transform coding is another example of the transform coding technique. Sinusoidal transform based coders [15-17] have been developed over the past few years, and have shown an ability to produce good quality speech at rates from 2.4kb/s to 4.8kb/s.

Currently, the CELP algorithm is the basis for many speech coding standards. The main issue is the demand for speech quality versus bit rate. In low bit rate speech coding, this requirement appears to be met by the CELP algorithm [18]. At a rate of 4.8kb/s the CELP algorithm can provide near-toll quality speech. However, as the bit rate is lowered, the coded speech becomes poor and unnatural. The main reason for this is the lack of natural periodicity of the excitation, which is derived from an adaptive codebook and from a Gaussian codebook. New approaches for low bit rate coding have thus been concentrated in two areas [18]:

- enhancing the CELP algorithm, and
- creating new algorithms.

In the last few years there have been a number of contributions to both areas. In the first area, research has mainly concentrated on increasing the degree of periodicity of excitation by separating voiced and unvoiced codebooks. A voiced codebook is designed as a trained codebook or an impulsive codebook for voiced frames, while in unvoiced frames a Gaussian codebook is used [19-22].

A major contribution to the second area has come from the research on extracting prototype waveforms and interpolating between them [23]. Such an

approach can overcome the problems of CELP and can produce high quality speech at very low rates. Unlike ordinary waveform coders (either transform or sub-band coders), prototype waveform coders represent input speech (or the residual) as a series of prototype waveforms. In other words, speech (or the residual) can be described as a two-dimensional signal; one axis is time where the prototype waveform evolves and the other describes the prototype waveform shape.

This thesis investigates the prototype waveform coding approach. The aim is to find speech coding algorithms at rates of 2.4kb/s which can produce coded speech with quality at least equal to that of the 4.8kb/s CELP algorithm.

1.2 Contributions and Publications

The original contributions of this thesis can be briefly described as follows:

1) A new algorithm for Multi-Prototype Waveform (MPW) coding in an open loop architecture (MPW Open Loop Quantisation). The algorithm is based on exploiting the natural periodic property of the speech, and the prototype waveform decomposition. Two different MPW coders operating at 2.4kb/s have been developed as a result of this algorithm (Chapter 4).

2) A new algorithm for Multi-Prototype Waveform coding in an analysis-by-synthesis architecture (MPW Analysis-by-Synthesis Quantisation). The algorithm makes improvements to the MPW Open Loop Quantisation. On the basis of this algorithm four different 2.4kb/s MPW Analysis-by-Synthesis coders have been developed (Chapter 5).

3) A codebook solution for quantising prototype waveforms is introduced. Based on this, 8 bit SEW codebooks and 5 bit REW/*Error* codebooks have been designed (Chapter 4).

Publications arising from the work described in this thesis are as follows:

1. D.H. Pham and I.S. Burnett, "Quantisation Techniques for Prototype Waveforms", *IEEE International Symposium on Signal Processing and its Applications*, Gold Coast, Australia, 1996.

2. A paper preliminarily entitled: "Analysis-by-Synthesis Quantisation Techniques for Prototype Waveforms", is also being submitted to the *IEEE International Conference on Acoustic, Speech and Signal Processing*, Munich, Germany, 1997.

1.3 Organisation of the Thesis

This thesis is divided into five chapters. Chapter 2 reviews current low bit rate speech issues and provides an introduction to the work of this thesis. Chapter 3 presents the basics of Prototype Waveform Coding. Chapter 4 proposes two prototype waveform quantisation schemes in an open loop architecture operating at bit rates of 2.4 kb/s. The chapter also describes the codebook solution design for prototype waveforms and Line Spectral Frequencies. Chapter 5 presents a second coding algorithm using an analysis-by-synthesis technique. The operation as well as the advantages of this technique over the open loop quantisation technique are discussed. The experimental tests of the first and the second coding algorithm are performed using the Mean Opinion Score criterion (MOS) and compared with the US 1016 Federal Standard CELP-4.8kb/s. These tests are presented in Chapters 4 and 5 respectively. Finally, Chapter 6 presents conclusions drawn from the work described in this thesis. As a result, possible future work is suggested. This is expected to lead to further improvements in prototype waveform coding algorithms.

Chapter 2: Literature Review

This chapter discusses the main issues and design aspects of the speech coding techniques presented in this thesis. As mentioned in the previous chapter, many current low bit rate speech coding applications use CELP based algorithms. It is, thus, worth reviewing this speech coding algorithm to highlight critical issues in current speech coding techniques. Consequently, the various approaches of documented prototype waveform coding techniques are presented. Finally some aspects such as vector quantisation and quality measurements for speech coding are considered.

2.1 CELP Coding Technique

Code-Excited Linear Prediction (CELP) has been a widely used speech coding technique since the late 1980s. Good quality speech can be obtained at bit rates above 4.8kb/s, however, as the bit rates are lowered the speech quality degrades rapidly.

In CELP, the speech is coded with an analysis-by-synthesis procedure of speech waveform matching on a frame-by-frame basis. Exploiting the masking properties of human hearing, CELP uses a perceptual weighting function for

improving subjective quality. The excitation for the synthesis filter is derived as a sum of two gain-scaled vectors [2]. One vector is chosen from an adaptive codebook which contains the past excitation [24]. Thus the adaptive codebook vector produces long term periodicity by repeating sections of the past excitation for the present sub-frame. The second vector is taken from a fixed stochastic codebook. The codebook searching algorithm for the two vector codebooks is performed by minimising the perceptually-weighted error between the original and reconstructed speech [2]. The basic structure of the US 1016 Federal Standard CELP-4.8kb/s [7,8] can be described by the block diagram shown in Figure 2.1.

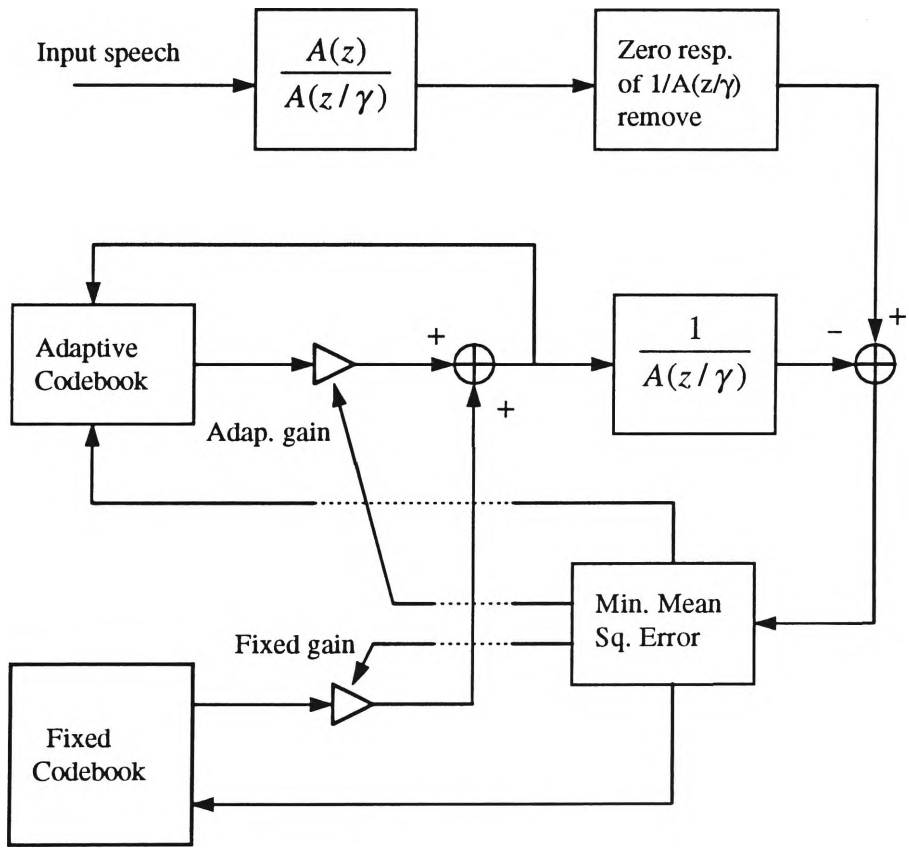


Figure 2.1 Standard CELP Architecture (based on [7,8])

The components of the coder are an LPC analysis filter, $A(z)$, a perceptually-weighted LPC synthesis filter, $1/A(z/\gamma)$, an adaptive codebook, and a

stochastic fixed codebook. The Federal Standard 1016 Coder uses 30ms frames containing 240 speech samples at a sampling rate of 8kHz. For each frame, the 10^{th} order LPC parameters are estimated using the autocorrelation method (either the Levinson-Durbin [25] or the Schur [26] recursion algorithm). These parameters are coded using Line Spectral Frequencies (LSFs) [27]. The frames are divided into 4 sub-frames, each of length 60 samples (or 7.5ms). For every sub-frame both adaptive and fixed codebook search are performed. The adaptive codebook's length is 256 vectors (corresponding to 128 noninteger delays and 128 integer delays), and the fixed codebook contains 512 Gaussian codewords with an overlap of 2 [28-30]. The adaptive codebook contains the history of the residual and can be regarded as a sample based shift storage register [31].

A bit allocation for the Federal Standard 1016 CELP coder [29,32] is presented in Table 2.1. In this scheme, 138 bits, plus a number of other supplementary bits (1 bit for synchronisation, 4 bits for error correction, and 1 bit for future expansion) are needed. The total 144 bits per 30ms frame is equivalent to 4.8kb/s.

Parameter	Bits/subframe	Bits/frame
LSFs (FS 1016 scheme)	-	34
Adaptive codebook index	(8+6+8+6)	28
Adaptive codebook gain	5 (4)	20
Fixed codebook index	9 (4)	36
Fixed codebook gain	5 (4)	20
Total bits	26 (4)	138

Table 2.1 Bit Allocation for the US 1016 Federal Standard CELP-4.8kb/s

In alternative CELP schemes, the bit rate is varied by varying the frame size, the codebook size, or changing the bit allocation plan of the coder. Good speech quality can be obtained at bit rates above 4.8kb/s, but, as the bit rate is lowered, the speech quality as well as its naturalness decrease rapidly [33]. For lower bit rate coding, the frame and sub-frame sizes are larger, the stochastic codebook size is smaller and the adaptive and fixed gains are quantised more coarsely. This leads to a reduction in the ability to produce periodic excitation in the coders. As the fixed codebook size is reduced the fluctuations in the spectrum of the coded speech increase [22].

In short, the problems of CELP at low rates are mainly caused by the increasing inaccuracy in the waveform matching and the lack of an accurate degree of periodicity in the voiced speech signal [34].

2.2 Prototype Waveform Coding Technique

Due to the limitation of CELP coding at low bit rates, there has been much interest in finding alternative coding algorithms. In recent years, several techniques have been proposed. Prototype waveform coding is one technique. It exploits the periodic property of the speech signal by extracting pitch cycle waveforms and interpolating between them. This approach leads to two main issues:

- how to code the extracted pitch cycle waveforms, and
- how to interpolate them.

While the second issue is the basis for prototype waveform coding, the first issue can be considered as the key to achieving high quality speech at low bit rates. New prototype waveform based coding methods were implemented in the time domain, the frequency domain or the mixed domain. Generally, they

can be classified into two different coding modes: multi-mode coding and single mode coding.

2.2.1 Multi-Mode Coding

In this speech coding method, the periodicity of the speech signal was exploited to code the voiced speech, while unvoiced speech was coded using another technique such as CELP, employing only a fixed codebook search.

The first significant publication on prototype waveform coding was the technique proposed by Kleijn [34]. In this paper, the author described voiced speech as a quasi-periodic signal which is a concatenation of pitch cycle waveforms or prototype waveforms (PWs). Voiced speech signals can be coded at low bit rates by extracting and coding the PWs. At the receiver, the speech can be recovered by continuous interpolation between these PWs. Based on this principle, he proposed a technique called Prototype Waveform Interpolation (PWI). The coding algorithm operated in the DFT domain because of certain computational advantages over time domain techniques. The proposed coders [34-36] coded voiced and unvoiced speech signals separately. The voiced speech was coded by the PWI technique and the unvoiced speech using the CELP algorithm. Since the PWs are slowly evolving signals, it is possible to downsample to one PW per frame of 20-30ms, and hence low bit rate coding can be achieved. The downsampled PWs can be quantised using vector quantisation. At the decoder, the voiced speech signal can be reconstructed by interpolation from a sequence of the quantised PWs. This coder was reported to be capable of producing excellent voiced speech quality at rates between 3.0 and 4.0kb/s.

Based on the PWI technique [34], Burnett and Holbeche [37] developed a low bit rate coder wherein different quantisation techniques for the prototype waveform were investigated. Their coder also quantised voiced speech using

the PWI technique, and unvoiced speech using CELP without an adaptive codebook search.

High complexity due to the use of stochastic codebooks and the DFT domain operation is one of the significant drawbacks of the coder proposed by Kleijn [34-36]. Consistent with the technique proposed by Kleijn [34], Yang, et.al. [33] proposed a coding method which employs the majority of the bits to encode a part of the voiced waveform and uses a forward and backward waveform prediction technique to reconstruct the complete voiced signal at the decoder. In an attempt to overcome the complexity of Kleijn's coder, this coder was designed to operate in the time domain. This technique was only used for coding voiced speech frames. Unvoiced frames were coded using the CELP algorithm (again, without an adaptive codebook search). In fact, this algorithm did not extract the whole pitch cycle waveform for coding, it exploited the periodicity of the speech signal for both backward and forward prediction. Partial voiced speech waveforms in each frame are encoded, and the complete voiced speech waveform of the whole frame recovered by means of backward and forward prediction (i.e., waveform interpolation). The method was reported to be capable of producing high quality voiced speech at rates between 3.0 and 4.0kb/s. In terms of complexity, this method had advantages in comparison with those operating in the frequency domain. However, in this method it is easy to have discontinuities at boundary points of the prediction.

Tang and Cheetham [38] described a method for low bit rate speech coding based on the PWI technique proposed by Kleijn [34]. Aimed at reducing the computational complexity, this paper suggested a prototype waveform coding technique using variable frame lengths. Each frame can contain an integer number of pitch periods. As a result, the time alignment between prototype waveforms can be eliminated, and the computational complexity can be

reduced. However, control of the bit rate of the coder proved to be a serious problem. Further work on this technique has remained unreported.

Tanaka and Kimura [39] introduced a speech coding technique using a two dimensional Fourier transform to code pitch waveforms of the residual signal. This technique was aimed at solving a problem with Kleijn's technique [34], in that the coded speech can deteriorate when using a long interpolation frame or a short pitch waveform. The basic idea of the new technique was that each frame of voiced speech residual should be regarded as a sequence of pitch waveforms. By circular shifting and zero padding, the pitch waveforms are maximally aligned, and have the same length. Then the residual frame can be considered as a matrix of these pitch waveforms. A Fourier transform of the matrix was performed by means of a one-dimensional transform on the columns and the rows sequentially. Only the low transition frequency components were transmitted. Because of the periodic property of the voiced residual signal, most of the energy is concentrated in the low transitional frequency band in the two-dimensional transform domain [39]. Due to the high computational and memory requirements of this technique, Tanaka and Kimura proposed an alternative method which involved a combination the two dimensional Fourier transform and waveform interpolation. Furthermore, they also introduced a multi-band approach to the two-dimensional transformation technique. These proposed techniques were also used only for voiced speech, and were combined with CELP for unvoiced speech.

Shoham [18] suggested a speech coding technique at bit rates between 2.4 and 4.0kb/s based on time frequency interpolation (TFI). The technique coded voiced speech and unvoiced speech separately. Similar to the coding techniques discussed previously, voiced speech was coded using the time frequency interpolation technique, while CELP was used for coding unvoiced speech. The time-frequency interpolation scheme exploited the periodic

feature of the speech to apply the DFT to approximately pitch size segments of the residual signal. Magnitude and phase spectra of the DFT signal were separated, and then vector quantised using a weighted variable-size, multi-stage, predictive vector quantiser. This algorithm generates good quality speech, however, it is empirical [18] and the coding algorithm is sophisticated.

In these techniques, reportedly, high voiced speech quality can be obtained at bit rates less than 4.0kb/s, however, the voiced and unvoiced speech were coded using different mechanisms. The unvoiced speech was still coded by the CELP algorithm without an adaptive codebook. This coding method may lead to discontinuities in the reconstructed speech at the boundaries between voiced sections and unvoiced sections [40,41].

2.2.2 Single Mode Coding

For higher efficiency prototype waveform coding, Kleijn and Haagen [42] introduced a new waveform interpolation method which extracts prototype waveforms at a high rate during both voiced and unvoiced speech. In this algorithm, the speech signal is considered as a sequence of Characteristic Waveforms (CWs) or Prototype Waveforms (PWs). The characteristic waveform is decomposed into a Slowly Evolving Waveform (SEW) and a Rapidly Evolving Waveform (REW). The SEW represents quasi periodic components of speech, and dominates during voiced speech segments. The REW is noise-like, and dominates during unvoiced speech segments. Because of their different properties, the two PW components, the REW and SEW, can be quantised separately and a high coding efficiency can be obtained.

On the basis of this algorithm, Kleijn and Haagen [40,43] proposed a Prototype Waveform coder operating at a bit rate of 2.4kb/s. Basically, the coder extracted the PWs and interpolated between them in the DFT domain. The extracted PWs (at an update rate of 480Hz) were normalised and

decomposed into a SEW and REW by means of a low-pass filter and high-pass filter respectively. The SEWs were downsampled to 40Hz (i.e., one SEW per frame). The phase and magnitude spectra of the SEW were separated for quantisation. The coder employed a 7 bit vector quantiser to quantise the SEW magnitude spectra. The SEW phase spectra is not quantised, but it is inferred from the transmitted REW parameters [43]. The REWs were downsampled to 240Hz (i.e., 6 REW per frame) and then separated into REW phase and REW magnitude spectra. The REW magnitude spectra can be quantised using 3 bits (8 shapes). The REW phase spectra is noise-like, so it was not quantised. At the receiver, the complete PW was recovered using the transmitted REW and SEW parameters in combination with a new random phase for each REW and based on the assumption that the magnitude spectra of a normalised PW is approximately flat and unity. The performance of this coder was at least equivalent to the 4.8kb/s US 1016 Federal Standard [43]. It should be noted that this work was published concurrently with the work being undertaken in this thesis.

Consistent with the algorithm proposed by Kleijn and Haagen [42], Burnett and Bradley [41] suggested a Multi-Prototype Waveform (MPW) coding technique. In this technique, ten prototype waveforms (PWs) were extracted every frame. The extracted PWs were then normalised and decomposed into the SEW and the REW in a similar way to that described by Kleijn. In this scheme, the REW was quantised with either an open-loop or a closed loop architecture [41]. The SEW was not quantised (a simple model was used) and at the receiver, the PW was recovered by using the transmitted REW to reconstruct the complete PW. At a bit rate of 2.84kb/s this coder was reported to be capable of producing good quality speech. In this method, these authors also investigated defining the SEW in a slightly different way to that used by Kleijn and Haagen [42], such that it was considered as a 'mean' PW of the ten extracted PWs in every frame [41,44].

Due to certain complexities in the coder reported in [40] and [43], Kleijn, et.al. introduced a new prototype waveform coder [45] with less complexity. The new prototype waveform coder was similar to the former, but it had a number of new features for complexity reduction and speech quality improvement. These features: Spectral Colouring and Fast Synthesis using Cubic Splines, were included in the decoder. The Spectral Colouring was performed by multiplying the DFT coefficients with the combined transfer function of the all-pole LP synthesis filter and the pole-zero postfilter. This publication was concurrent with the time this thesis was being completed.

2.3 Thesis Objectives

Exploiting the periodic property of the speech signal by extracting prototype waveforms (PWs) and smoothly interpolating between them is currently proving a suitable method for coding speech at low bit rates. The decomposition of PWs into a slowly evolving waveform (SEW) and a rapidly evolving waveform (REW) has been found to be an effective method for quantising the PWs. This method can be used for coding voiced, unvoiced speech and also background noise [42]. It is, thus, an improvement over the multi-mode coders. Another idea, the definition of the SEW as a mean prototype waveform [41,44] is one of the areas being investigated in this thesis.

Having investigated the literature, this thesis researches low bit rate speech coding methods based on exploiting the periodic property of speech and the REW/SEW decomposition paradigm [41,42]. As previously discussed, there are two issues in prototype waveform coding techniques: how to quantise prototype waveforms and how to interpolate between them. As reported in the literature [34-37,40-45], although there is a certain computational complexity, interpolation of the PWs is still best performed in the DFT domain. This thesis

uses the results obtained and concentrates on quantisation techniques for prototype waveforms. Since the majority of operations are performed in the DFT domain, quantisation is also examined in that domain.

In the DFT domain, or generally in the transform domain, it is effective to quantise the transform coefficients by vector quantisation (VQ) rather than scalar quantisation (SQ) [46]. Further, with a fixed coding rate, the performance of VQ is always better than that of SQ [47]. Chang, et.al. [46] also show that in the transform domain each sample depends on many other samples in the original domain, thus VQ can provide better performance than that of SQ. Moreover, VQ [52] of the speech waveform in the DFT domain provides distinctively better subjective quality than VQ of the speech waveform in the original domain [46]. Because of these advantages, this thesis uses VQ for the purpose of coding the prototype waveforms at low bit rates. The vector quantisation techniques are now reviewed.

2.4 Vector Quantisation

This section considers the main issues in VQ techniques which will be related to the prototype waveform quantisation and the quantisation of other parameters such as Line Spectrum Frequencies.

VQ [48,49] is a quantisation process wherein a comparison between an input signal block of pre-determined size (a vector) and a pre-determined set of vectors (a codebook) is performed. The comparison can be in the form of a Mean Squared Error measure, or Weighted Mean Squared Error measure [50]. The best matching codebook vector is chosen and its associated index is transmitted or stored. At the receiver, the block of signal is reconstructed by substituting the transmitted codebook index with its codebook vector.

2.4.1 Description of Vector Quantisation

In VQ, a data vector of N dimensions $\mathbf{x} = \{x_k,\ 1 \le k \le N\}$ is mapped onto an N dimensional vector $\mathbf{y}_i = \{y_{ik},\ 1 \le k \le N\}$ of a pre-determined set $\mathbf{y} = \{\mathbf{y}_i,\ 1 \le i \le L\}$. $\mathbf{y}$ is termed the codebook which contains L codewords of N dimensions. This quantisation can be considered as the N dimensional space, characterised by the N dimensional vectors $\mathbf{x}$, being partitioned into L cells $\{C_i,\ 1 \le i \le L\}$. Each codeword in the codebook represents one of these cells. A data is quantised by the codeword of the cell into which it falls.

$$q(\mathbf{x}) = \mathbf{y}_i \qquad \text{if } \mathbf{x} \in C_i. \tag{2.1}$$

For the case $N = 1$, vector quantisation becomes scalar quantisation. In other words, scalar quantisation is a special case of vector quantisation.

2.4.2 Distortion Measures

There are many distortion measures proposed in the literature, but the most common for convenience of mathematics is the Mean Squared Error (MSE) [51]. If the distortion caused by the input vector $\mathbf{x}$ and the quantised vector $\mathbf{y}$ is denoted as $d(\mathbf{x}, \mathbf{y})$,

$$d(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{N} \|x_k - y_k\|^2 . \tag{2.2}$$

Another mean squared error measure is the Weighted Mean Squared Error (WMSE):

$$d(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{N} w_k |x_k - y_k|^2 \tag{2.3}$$

where w_k is a weighting function which is dependent on the input vector. Several distortion measures other than those above, such as Linear Prediction

Distortion Measure, Perceptually Motivated Distortion Measure [52], have also been proposed since the 1980s.

2.4.3 Performance of Vector Quantisation

Shannon's source coding theorem with a distortion criterion has shown that VQ always achieves a better performance than SQ, even when the data is memoryless [53]. In fact, performance of a VQ will approach the rate distortion limit when the vector lengths tend to infinity [52]. However, VQ can achieve optimal performance for fixed codebook size and for very large codeword lengths [52]. This property enables high quality coding at low bit rates.

To estimate the performance of a VQ, several tools can be used such as the rate distortion theory [52-54]. Linde, et.al. [51] used a performance measure related to codebook design. This performance measure can be summarised as:

Let $\mathbf{y}$ be a VQ (a codebook) which contains L codewords of dimension N, $\mathbf{y} = \{\mathbf{y}_i, 1 \le i \le L\}$ with $\mathbf{y}_i = \{y_{ik}, 1 \le k \le N\}$. Let $\mathbf{z} = \{\mathbf{z}_k, 1 \le k \le N\}$ be a real random vector described by a cumulative distribution function $F(z) = \Pr\{\mathbf{z}_k \le z_k;\ 1 \le k \le N\}$. Applying the VQ $\mathbf{y}$ to the random vector $\mathbf{z}$, the performance $D(q)$ of this quantisation is measured as the expected distortion:

$$D(q) = Ed(\mathbf{z}, q(\mathbf{z})) \tag{2.4}$$

where E denotes the expectation with respect to the underlying distribution F [51]. Let a set of data be $\mathbf{x} = \{\mathbf{x}_n,\ n = 0,..., M\}$, with $\mathbf{x}_n = \{x_{nk},\ 1 \le k \le N\}\}$, that is, a sequence of stationary and ergodic vectors. The quantisation performance of VQ $\mathbf{y}$ when applying to the set of data $\mathbf{x}$ will be found as $D(q)$ if:

$$\lim_{M\to\infty} \frac{1}{M}\sum_{n=0}^{M} d(\mathbf{x}_n, q(\mathbf{x}_n)) = D(q) \quad \text{with a probability of one.} \tag{2.5}$$

Thus, $D(q)$ is regarded as the long run time averaged distortion.

2.4.4 Codebook Design

A quantiser is considered as an optimal (minimum distortion) quantiser if the two conditions below are satisfied [52].

The first condition is regarded as the nearest neighbour selection rule, whereby the choice of a codeword for an input vector is a result of the minimum distortion out of all other codewords in the codebook.

$$q(\mathbf{x}) = \mathbf{y}_i \quad \text{if } d(\mathbf{x}, \mathbf{y}_i) \le d(\mathbf{x}, \mathbf{y}_j) \quad \text{for } j \ne i;\ 1 \le j \le L. \tag{2.6}$$

The second condition is that the choice of codeword $\mathbf{y}_i$ is such that the average distortion in quantising the input vector falling in the cell C_i is minimised. That is, $\mathbf{y}_i$ is chosen so as to minimise:

$$D_i = \sum d(\mathbf{x}, \mathbf{y}_i) \qquad \mathbf{x} \in C_i. \tag{2.7}$$

Therefore, $\mathbf{y}_i$ is termed centroid of the cell C_i and is dependent on the distortion measure.

2.4.4.1 LBG Codebook Design

Linde, Buzo and Gray [51] extended Lloyds iterative design for a scalar quantiser to an algorithm for a vector quantiser, lately named as the LBG algorithm. This algorithm has been widely used in the codebook design because of its simple and effective properties.

Applying the two optimality criteria conditions above, the LBG algorithm produces a codebook based on a set of training data. The codebook training procedure is a process of iterations of four training steps. Details of the procedure can be found in the original paper, however, the basis of it is described as follows:

(1) *Initialisation*: Choose an initial codebook by a certain method: $\mathbf{y} = \{\mathbf{y}_i,\ 1 \le i \le L\}$ with $\mathbf{y}_i = \{y_{ik},\ 1 \le k \le N\}$. Set the distortion threshold ε.

(2) *Classification*: the set of training data of length M, are classified into L groups of M_i vectors falling in the same cell C_i, by using the nearest neighbour rule (applying Condition 1),

$$\mathbf{x} \in C_i \quad \text{if } d(\mathbf{x}, \mathbf{y}_i) \le d(\mathbf{x}, \mathbf{y}_j) \quad j \ne i. \tag{2.8}$$

(3) *Codebook Updating*: Compute the centroid of the training vectors in each group. The new centroids now form the new codebook. This step is performed by using Condition 2.

$$\mathbf{y}_i = \text{cent}(C_i) = \frac{1}{M_i} \sum_{\mathbf{x} \in C_i} \mathbf{x} \qquad 1 \le i \le L. \tag{2.9}$$

(4) *Checking for termination*: The process is repeated from Step 2 until the reduction in the overall distortion (compared with the overall distortion at the previous iteration) is less than or equal to the distortion threshold ε.

The overall average distortion measure can be computed as:

$$D(\mathbf{x}, \mathbf{y}) = \frac{1}{M} \sum_{i=1}^{L} \sum_{\mathbf{x} \in C_i} d(\mathbf{x}, \mathbf{y}_i). \tag{2.10}$$

This codebook design always guarantees non-increasing overall average distortion at each iterations of the training process. Thus, it can provide reliable codebooks for many coding applications.

Initial codebook: An initial codebook can be designed using several techniques. The simplest way is to randomly use a part of the training sequence as the initial codebook. Alternatively, an initial codebook can be designed by means of the splitting technique. Let the codebook needed to be trained have L vectors with length of N, this technique is described as follows [51,52]:

(1) Set the initial codebook $\mathbf{y}(0)$ containing l vectors $\{\mathbf{y}_i;\ 1 \le i \le l\}$ each has length of N: $\mathbf{y}_i = \{y_{ik},\ 1 \le k \le N\}$. Note that $l<L$. Split each vector $\mathbf{y}_i$ into two closed vectors $\mathbf{y}_i + e$ and $\mathbf{y}_i - e$, where e is a fixed perturbation vector. The new codebook $\tilde{\mathbf{y}}(0)$ contains $2l$ vectors $\{\mathbf{y}_i + e,\ \mathbf{y}_i - e,\ 1 \le i \le l\}$.
Replace l by $2l$. Continue this work until $l = L$.

(2) If $l = L$ set $\mathbf{y}(0) = \tilde{\mathbf{y}}(0)$ and halt. $\mathbf{y}(0)$ is then the initial codebook for the N level quantisation algorithm.

The choice of the vector e and the value l depends on each vector quantiser such as the LSF vector quantiser, SEW vector quantiser. Further details about this algorithm can be found in the original papers.

2.4.4.2 Codebook Design using Simulated Annealing

The codebook design technique introduced by Flanagan, et.al. [55] is based on Simulated Annealing (SA). Like the LBG, the SA algorithm uses Conditions 1 and 2 as a characterisation.

In the SA algorithm, a set of M training data $\mathbf{x} = \{\mathbf{x}_j;\ 1 \le j \le M\}$ wherein $\mathbf{x}_j = \{x_{jk};\ 1 \le k \le N\}$ is partitioned into L subsets $S = \{S_i;\ 1 \le i \le L\}$. Let G be an assignment of training data to the subsets S and the codebook be denoted as $\mathbf{y} = \{\mathbf{y}_i;\ 1 \le i \le L\}$ with $\mathbf{y}_i = \{y_{ik};\ 1 \le k \le N\}$. The average distortion now is defined as:

$$D(\mathbf{x}, \mathbf{y}, G) = \frac{1}{M} \sum_{i=1}^{L} \sum_{\mathbf{x} \in S_i} d(\mathbf{x}, \mathbf{y}_i) \qquad (2.11)$$

where $d(\mathbf{x}, \mathbf{y}_i) = |\mathbf{x} - \mathbf{y}_i|^2$. This distortion is not only a function of the training data and the codebook but also of the assignment of training data to subsets. Flanagan, et al. [55] showed that if $D(\mathbf{x}, \mathbf{y}, G)$ is minimised, $D(\mathbf{x}, \mathbf{y})$ in Equation (2.10) is also minimised, and thus an optimum codebook is obtained. Note that in the assignment G, the selected vector does not necessarily have to be closest to the code vector. Here an initial codebook is created by partitioning the set $\mathbf{x}$ of training data into L subsets $S_1 \ldots S_L$. Centroids $\mathbf{y}_1 \ldots \mathbf{y}_L$ of the initial codebook are calculated from the data vectors partitioned in each subset. A summary of the SA algorithm is presented below.

(1) *Initialisation*: Set a distortion threshold ε, repetition index k = 0, initial temperature parameter T_0 and the optimising control function f(.). The assignment G is performed by partitioning the training set $\mathbf{x}$ into L subsets $(S_1, \ldots, S_L)$ in a round bin or random fashion. The centroids of the initial codebook are formed from the elements in each subset (using Condition 2), and is defined as:

$$\mathbf{y}_i = \frac{1}{M_i} \sum_{\mathbf{x} \in S_i} \mathbf{x} \qquad (2.12)$$

where M_j is the number of elements in S_i.

(2) *Codebook Updating*: Randomly select a training vector and move it from its current subset into a new subset to form a new assignment G'. This perturbation of the codebook is completed by calculating the new centroids of each new subset according to (2.12) to obtain a new codebook $\mathbf{y}' = \{\mathbf{y}_i;\ 1 \le i \le L\}$.

(3) *Codebook Checking*: The change in distortion between the old and the new codebook is computed:

$$\Delta D = D(\mathbf{x}, \mathbf{y}', G') - D(\mathbf{x}, \mathbf{y}, G). \tag{2.13}$$

The perturbation is accepted if

$$e^{\Delta D/T_k} > r \tag{2.14}$$

where r is a uniformly distributed random number between 0 and 1.

(4) *Termination*: The process will be repeated from Step 2 by incrementing k such that $T_{k+1} = f(T_k)$ until the distortion is less than or equal to the distortion threshold.

This algorithm will converge if the following necessary and sufficient condition is satisfied [56]:

$$\lim_{K \to \infty} \sum_{k=0}^{K} T_k = \infty. \tag{2.15}$$

Hence, to converge to a certain value the algorithm requires a large number of iterations. The large number of iterations as well as the complexity of calculating ΔD is a problem with this technique. However, Flanagan, et al. [55] simplified the ΔD calculation such that the algorithm can be computed.

So far, two codebook design techniques have been discussed, however, the LBG algorithm has been widely used because of its simplicity and low complexity. In the work described in this thesis, the LBG is employed for designing codebooks for the prototype waveform quantisation. In general, the performance of a codebook is dependent on the length and the variety of the training data. In other words, the desired codebook should be trained using data that are representative of what the VQ meets in actual operation [52]. As the length of training data increases, the codebook performance improves.

However, Makhoul, et.al. [52] showed that if the length of the training data is 50 times the number of codebook vectors, it is considered sufficient for most applications. Sometimes, a ratio as few as 10 can be adequate. On the other hand, increasing the variety of training data (for example increasing the number of speakers in the training data) rather than increasing the amount of one sort of data (for example, the amount of speech from each speaker) can provide improved codebook performance [52].

2.4.5 Applications of VQ in Prototype Waveform Quantisation

Vector quantisation can be classified into different types such as memoryless VQ, feedback VQ, etc., [53]. There are a number of variations in both memoryless VQ and feedback VQ. For example, forms of memoryless VQ include: Tree-searched VQ, Multi-step VQ, Gain/shape VQ, Separating Mean VQ, Lattice VQ [53]. These all have certain advantages in particular applications. In this thesis, Gain/shape VQ and Separating Mean VQ are used for prototype waveform quantisation. The reasons for this choice are that the Gain/shape and Separating Mean VQ are suitable for quantising the PWs and, in terms of computational cost and subjective quality, these VQ have benefits over ordinary VQ.

2.4.5.1 Gain/shape VQ

In this quantisation technique, a speech waveform is separated into two interdependent parts: shape and gain [53]. The shape is regarded as the original input signal after normalisation. This can be considered to be the extraction of a gain term (power) from the input signal. The shape and the gain are quantised separately. The shape can be quantised by VQ, while the gain can be easily quantised by SQ. Quantising the PWs by the Gain/shape VQ

leads to better performance and yields smaller computational complexity than an ordinary VQ.

2.4.5.2 Separating Mean VQ

Similar to Gain/shape VQ, in the Separating Mean VQ [53] a sample mean instead of a gain term is extracted. After removing the sample mean, the original input signal becomes a new signal with approximately zero sample mean. The sample mean is then quantised using SQ. The new signal can now be quantised as the difference between the original input signal and the sample mean using VQ.

2.5 Quality Measurements for Speech Coding

In speech coding, the measurements for coded speech quality are divided into two classes: Objective and Subjective Measurements [57,58]. Objective Measurements are easily conducted by mathematical formula calculations, while Subjective Measurements require more time to be carried out since they are based on the opinions of listeners. The human ear is sensitive to certain forms of distortion, but some others cannot be perceived because of masking and threshold effects [31]. A small quantisation error does not mean that the distortion in the speech signal is perceptually small [59]. When the periodicity of voiced speech of CELP increases, the perceptual quality of speech increases although the value of the objective measure decreases [60]. Thus, the use of objective and subjective measurements in speech coding is dependent on the individual coding algorithm. As with many prototype waveform coders, the prototype waveform coders described in this thesis are tested using subjective measurements. Objective and subjective measurements are discussed in the following sections.

2.5.1 Objective Measurements

There are three standard objective measures which are widely used in speech coding: Average SNR, Segmental SNR and Cepstral Distance.

The Average SNR [58] is defined as the average value of a large number of frame SNRs. Each frame SNR is defined as the ratio between the input speech and the error between the input speech and the output speech. If $x(n)$ and $y(n)$ are defined as the input speech and output speech, the average SNR is defined by:

$$Av.SNR = 10\log_{10}\left[\frac{1}{N}\sum_{f=1}^{N}\frac{\sum_{n=0}^{L-1}x^2(n)}{\sum_{n=0}^{L-1}[x(n)-y(n)]^2}\right] \tag{2.16}$$

where N is the number of frames and L is the length of those frames.

The Segmental SNR [58,61] is defined as:

$$Seg.SNR = \frac{1}{N}\sum_{f=1}^{N}10\log_{10}\frac{\sum_{n=0}^{L-1}x^2(n)}{\sum_{n=0}^{L}[x(n)-y(n)]^2} \tag{2.17}$$

Cepstral Distance differs from the previous measures, in that it is a spectral distortion measure. There are several methods used to evaluate Cepstral Distortion. Further details of Cepstral Distortion can be found in [61-64].

Unfortunately, objective measures are not generally applicable to prototype waveform coders since the input and output waveforms are not time synchronous.

2.5.2 Subjective Measurements

Subjective measures consist of several different classes. However, the most widely used is the Mean Opinion Score (MOS) [57,58]. This measure requires a large number of listeners to classify synthesised speech on a five point scale as shown in Table 2.2. For reliable results, reference speech, including many speakers with different accents, is required to be evaluated by a large number of trained listeners. The test is conducted in two phases [31]. Firstly, the listeners are trained during the test wherein they hear signals representing 'high', 'low', and 'middle' categories. The second phase is evaluation. In this phase, the listeners listen and classify the signal samples based on the table below. To achieve good test results, the listeners need to be well trained, that is, they have experience of the distortion forms and can make correct decisions of classification [65]. In the following chapters, subjective measures are used to evaluate the Multi-Prototype Waveform coders.

Scores	Speech Quality	Distortion Scale
5	Excellent	Imperceptible
4	Good	Just perceptible but not annoying
3	Fair	Perceptible and slightly annoying
2	Poor	Annoying but not objectionable
1	Unsatisfactory	Very annoying and objectionable

Table 2.2 Mean Opinion Score Standard [58]

2.6 Summary

This chapter has considered the main issues in low bit rate speech coding. Since the late 1980s the CELP technique has been used as the basis for many

speech coding standards. However, as the bit rate is decreased lower than 4.8kb/s, the coded speech quality degrades rapidly. This problem is mainly due to the lack of natural periodicity in the coded speech. At bit rates lower than 4.8kb/s, alternative, prototype waveform coding techniques have been proposed. Amongst them, prototype waveform coding is one class.

Basically, prototype waveform coding techniques exploit the periodic property of speech by extracting prototype waveforms (PWs) and interpolating between them. Coders which coded the voiced and unvoiced speech separately using a mixed technique of Prototype/CELP, reportedly, can improve coded speech quality over conventional CELP. However, this coding method was not optimal. Efficient speech coding can be obtained when PWs are extracted in both voiced and unvoiced speech and quantised using the SEW/REW paradigm. As the SEW and REW have certain distinctive properties, they can be quantised separately with different requirements. Thus, low bit rate coding can be achieved due to the dependence of bit allocation on these requirements. The prototype waveform coders based on this technique can produce high quality speech at bit rates as low as 2.4kb/s. Although there is a certain complexity, prototype waveform coding has been found to be best performed in the DFT domain.

As a result of the literature review, the work undertaken in this thesis focussed on finding new low bit rate prototype waveform coding algorithms which would make improvements to the SEW/REW paradigm.

Vector quantisation is one of the keys to achieving high quality speech in prototype waveform coding. VQ in the DFT domain has been found to be advantageous over either VQ in the time domain or SQ in the DFT domain. To obtain good codebooks for SEW/REW as well as for LSFs, the codebook training algorithm LBG proved useful. As the set of training data is longer, the codebook performance is better. However, the codebook performance also

depends on the variety of speech in the training set. Generally, if the length of the training set is 50 times the codebook size, it is sufficient to obtain a good codebook.

The speech quality can be assessed by objective or subjective quality measurements, of which Mean Opinion Score (MOS) is considered the best for assessing prototype waveform coders. However, difficulties with MOS are that it consumes time and depends on the listeners.

Chapter 3: Prototype Waveform Coding

This chapter presents the principles of Prototype Waveform Coding. The architecture and operation of a basic Multi-Prototype Waveform coding system is described. Techniques involved in this work, such as pitch detection, prototype waveform extraction, time alignment, prototype waveform continuous interpolation are discussed. For coding of speech to be effective, the prototype waveforms are required to be decomposed into distinct components, the quantisation requirements of which are different and dependent on their characteristics. This chapter also discusses decomposition paradigms for prototype waveforms.

3.1 Background

This section discusses the operational principles of a speech coder based on prototype waveform coding techniques, called the Multi-Prototype Waveform (MPW) coder. For the discussion to be useful, a definition of prototype waveforms as a description of speech is given initially.

3.1.1 Prototype Waveform

The term 'Characteristic Waveform' (CW) or 'Prototype Waveform' (PW) was first introduced by Kleijn in 1991 for voiced speech only [34]. Later, this concept was extended by Kleijn and Haagen in 1994 [42] for unvoiced speech. This extension allowed effective coding of both voiced and unvoiced speech at low bit rates. The definition of prototype waveforms is as follows:

Speech is a combination of voiced segments and unvoiced segments. During voiced segments, the speech is a quasi-periodic signal of pitch cycles. The pitch cycle waveform (or prototype waveform) evolves slowly with time. Voiced speech can thus be represented as a two-dimensional signal; one axis represents the evolution of the speech in time and the other the shape of each pitch cycle waveform. This concept, while natural for voiced speech, is also valid for unvoiced speech [40]. For the definition to be meaningful with both voiced and unvoiced speech, the term: 'pitch cycle waveform' is now replaced by the term: 'characteristic waveform' or 'prototype waveform'. During unvoiced speech, the prototype waveform evolves rapidly; its rate of change is a function of the periodic level of the speech signal.

This new concept suggested that the speech (or residual) signal can be represented as a series of prototype waveforms, the evolution of which is slow during voiced speech and rapid during unvoiced speech. The PWs can then be extracted and quantised. At the receiver, the synthesised speech can be obtained by smooth interpolation between the reconstructed PWs. In this work, the PW is extracted from the residual signal at a rate of 400Hz (i.e., 10 PWs per frame of 25ms) for both voiced and unvoiced speech. Examples of the residual signal represented as a series of PWs are described in Figure 3.1 and Figure 3.2. This residual is extracted from a segment of voiced speech spoken by a male speaker. In Figure 3.1 this representation is presented as a one-dimensional signal while Figure 3.2 presents it as a two-dimensional signal.

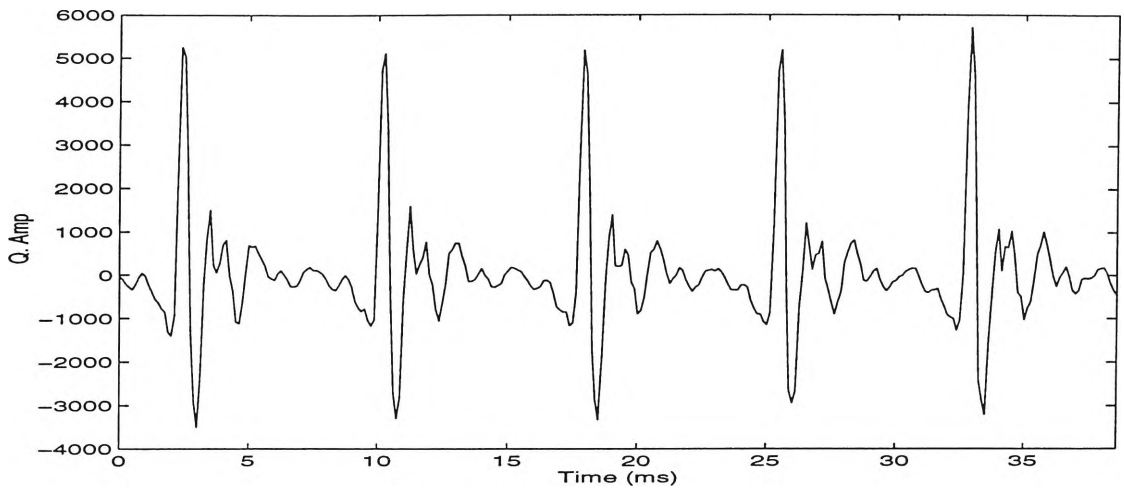


Figure 3.1 Residual Signal as a Series of Prototype Waveforms (One-dimensional Signal)

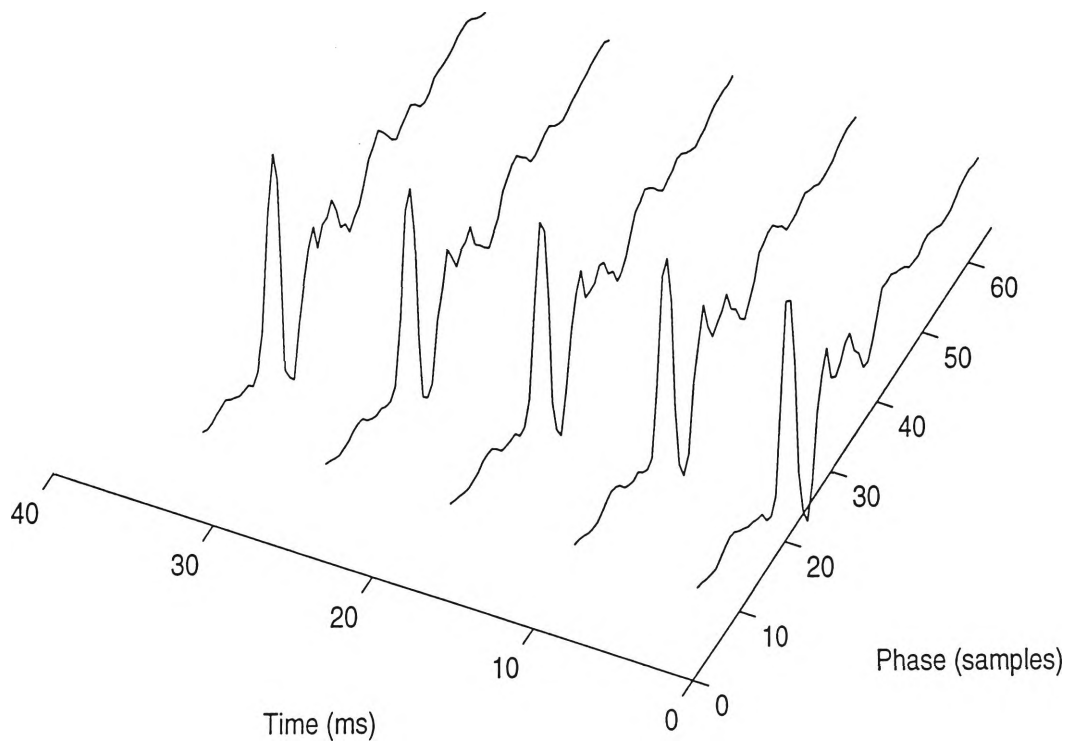


Figure 3.2 Residual Signal as a Series of Prototype Waveforms (Two-dimensional Signal: one axis is the evolution of signal in time and the other is the phase (or shape) of prototype waveforms)

3.1.2 Prototype Waveform Coder

The basic architecture of a MPW coder is described in Figure 3.3 with the upper section (A) being the encoder and the lower section (B), the decoder. This section discusses the encoding and decoding procedures of this coder.

Encoder

As in many coders, the first step of the encoding process is the estimation of the 10^{th} order LPC coefficients [25] for each 25ms frame (containing 200 samples at a sampling rate of 8kHz). This is implemented using the Schur recursion algorithm [26]. Once the LPC coefficients are obtained, the input speech is filtered to produce a LP residual signal by using a LP-analysis filter (a FIR filter) with the system function:

$$A(z) = 1 - \sum_{k=1}^{P} a(k) z^{-k} \tag{3.1}$$

such that the expression for analysis is:

$$r(n) = s(n) - \sum_{k=1}^{P} a(k) s(n-k) \tag{3.2}$$

where $r(n)$ and $s(n)$ are the residual signal and the input speech signal respectively.

For the filtering to be effective, the LP analysis filter is constructed in the form of a lattice structured filter [66]. This filter uses the reflection coefficients $(-k(i))$ as multipliers. These coefficients always have absolute values less than unity and are less spectrally sensitive to quantisation than the prediction coefficients $(a(i))$. As an example, a LP analysis filter and LP synthesis filter in the lattice structure are shown in Figure 3.4.

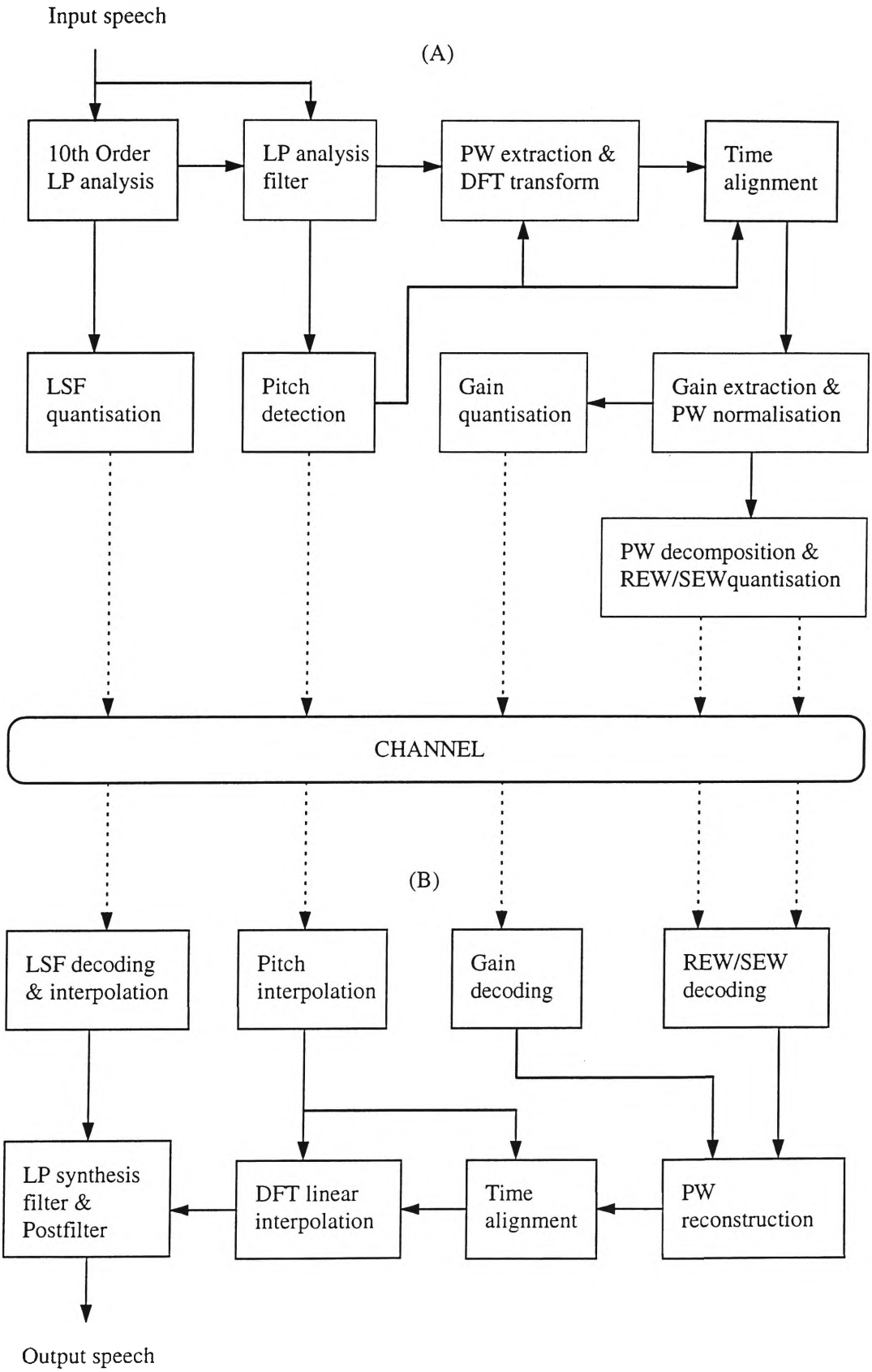


Figure 3.3 MPW Coder Architecture: (A) Encoder, (B) Decoder

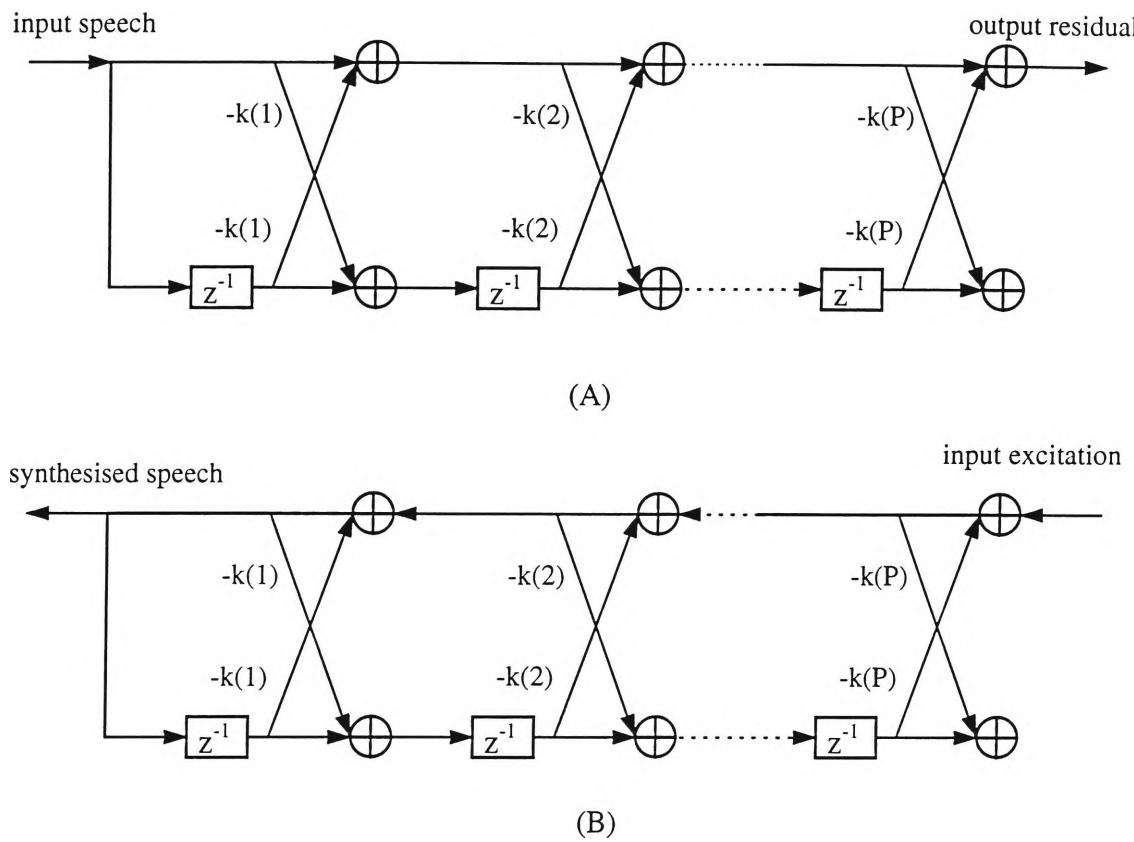


Figure 3.4 LP Lattice Filter Structures: (A) Analysis Filter, (B) Synthesis Filter

Reflection coefficients are used to guarantee the stability of the filters (either FIR or IIR). However, for the purpose of transmission at low bit rates over high error rate radio channels, these coefficients are not suitable. For the transmission to be less distorted in these environments, Line Spectral Frequencies (LSFs) are usually used instead of LPC coefficients. LSFs are defined by the following description: The system function of an all-pole (IIR) synthesis filter is defined as $H(z) = 1/A(z)$, where $A(z)$ is given by:

$$A(z) = 1 + a(1)z^{-1} + a(2)z^{-2} + a(3)z^{-3} + ...a(P)z^{-P} \qquad (3.3)$$

Line Spectral Frequencies can be determined by rewriting (3.3) in a new form:

$$A(z) = [F_1(z) + F_2(z)] / 2 \tag{3.4}$$

where the symmetrical polynomial $F_1(z)$ and the anti-symmetrical polynomial $F_2(z)$ relate to $A(z)$ as follows:

$$F_1(z) = A(z) + z^{-(P+1)} A(z^{-1}) \tag{3.5}$$

$$F_2(z) = A(z) - z^{-(P+1)} A(z^{-1}). \tag{3.6}$$

The roots of the two polynomials (3.5) and (3.6) determine the Line Spectral Frequencies. As a result of this definition, the LSFs gain two important properties of these polynomials:

- All zeros of $F_1(z)$ and $F_2(z)$ lie on the unit circle.
- Zeros of $F_1(z)$ and $F_2(z)$ are interleaved, thus the LSFs become interleaved.

These properties make LSFs more suitable for quantisation and transmission than LP coefficients. Details on LSFs, and the calculation of LSFs is to be found in [27].

In this work, LSF coefficients are quantised at an update rate of 40Hz using a 30 bit split-VQ. For each frame, 10 LSF coefficients are split into three sets in the form of a 3-3-4 combination [67]. The split 3-3-4 vector quantisation codes the LSFs as three vectors; the first vector consists of the first three LSFs, the second, the second three and the third, the remaining four. Each of the three LSF codebooks has 1024 vectors (10 bits). For quantisation, the LSF codebooks are fully searched, by minimising the mean squared error between the input LSFs and the LSF codebook vectors.

Pitch period (or fundamental frequency) is estimated from the residual signal once per frame (i.e., at an update rate of 40Hz). As extraction of a prototype waveform (PW) from either the speech or residual requires a reliable pitch detector, the pitch determination method has an important role in prototype

waveform coding. Many pitch estimation methods have been suggested such as the those reported in [68-70]. The pitch detector chosen for this work is an adaptation of the techniques reported in [71-73,37]. A brief description of the pitch determination technique is given in Section 3.2. The extracted pitch (expressed in samples) will be used as the PW length, on which the prototype waveform extraction, the time alignment and the DFT interpolation are dependent.

The PWs are extracted from the residual signal every 20 samples in both voiced and unvoiced speech (i.e., 10 PWs per frame). During voiced frames, the length of the PW is the pitch period, while in unvoiced frames it is chosen to be long enough to avoid buzziness [74]. Experimentally, it was chosen as 40 samples. The prototype waveform extraction process is performed on the basis of minimising the difference between two ends of the PW. The chosen PW is then transformed to the DFT domain. Time alignment is then performed to ensure that the selected PW aligns with the previous PW. This alignment prevents impulsive auditory distortion in the output speech produced by interpolation between PWs [37]. This work, however, leads to non-synchronisation between the input speech and the synthesised speech due to the loss of information about the absolute position of PWs. The time alignment technique used here is similar to that described by Kleijn [34].

For the purpose of gain/shape VQ [53], once the PWs are aligned, the gain term (i.e., the power) of each PW is calculated, and normalisation is performed. The gain term and the normalised PW are quantised separately. As the PW is normalised, its magnitude spectrum can be approximated to be unity.

For coding at bit rates as low as 2.4kb/s, the normalised PW is required to be decomposed into a number of distinct components. Such components can be quantised with different requirements according to their characteristics. One

accepted mechanism for PW decomposition is that proposed by Keijn and Haagen [42] whereby the PW is decomposed into a slowly evolving waveform (SEW) and a rapidly evolving waveform (REW). The SEW and the REW can be obtained by filtering using low-pass and high-pass filters with a suitable cut-off frequency of 20Hz.

In this work, an alternative REW/SEW paradigm described by Burnett and Bradley [41,44] is exploited due to its low decomposition complexity (see Section 3.6). In this paradigm, the SEW is defined as the 'mean' PW of ten extracted PWs each frame and the REW is then regarded as the noise-like remainder of the SEW and the extracted PW. As a result, the SEW is downsampled to 40Hz. For quantisation of the PW, this work introduces quantisation methods for the REW/SEW. At bit rates of 2.4kb/s, the SEW quantisation has been found to be effective by using an 8 bit VQ; and the REW quantisation requirements are examined by using two quantisation schemes: Unity Magnitude Quantisation and Errored Magnitude Quantisation.

The gain term of each PW is further processed before quantisation. Firstly, the logarithms of the ten gain terms of the ten PWs in each frame (i.e., update rate of 400Hz) are taken and these are downsampled to a certain rate according to the bit allocation plan of the coder. By conversing to the logarithmic form, the distance between the minimum and the maximum of the data to be quantised, is reduced. Hence, the quantisation errors can be minimised and the gain can be coded more accurately. The downsampled gain term (in the logarithmic domain) is then quantised using a 5 bit SQ.

Decoder

The first step in decoding is to decode the pitch. The number of DFT coefficients of the band limited signal is determined by the pitch period, thus it

is required for reconstruction of the SEW, REW and the PW. To obtain good quality speech the decoded pitch is interpolated by either the linear or non linear interpolation method. The pitch interpolation used in this work is an adaptation of that described by Kleijn and Haagen [40]. While details on this pitch interpolation process can be found in the original paper, a brief description is presented as follows:

Let C be the nearest integer to the ratio of the previous and the current pitch; while p_p, p_c, t_p, t_c are the previous pitch, the current pitch, previous update time, current update time respectively. If the current pitch (at time t_c) is larger than the previous pitch (at time t_p), the interpolated pitch are calculated as:

$$p_t = p_p + [p_c(t - t_p)] / [(t_c - t_p).C] \quad \text{if } t_p \le t < (t_c + t_p)/2 \tag{3.7}$$

$$p_t = C.p_p + p_c(t - t_p)/(t_c - t_p) \quad \text{if } (t_c + t_p)/2 \le t < t_c. \tag{3.8}$$

If the current pitch (at time t_c) is smaller than the previous pitch (at time t_p), the interpolated pitch will be calculated as:

$$p_t = C.p_p + p_c(t - t_p)/(t_c - t_p) \quad \text{if } t_p \le t < (t_c + t_p)/2 \tag{3.9}$$

$$p_t = p_p + [p_c(t - t_p)] / [(t_c - t_p).C] \quad \text{if } (t_c + t_p)/2 \le t < t_c. \tag{3.10}$$

The gain terms are decoded using table indices and the inverse logarithm taken to retrieve the linear values. Then, these gain terms are upsampled to 400Hz (i.e., 10 gains per frame) by means of interpolation. Experiments in this work show that linear interpolation provides good performance, however, further improvement can be obtained if a combination of linear interpolation and step-wise interpolation is used [40].

The SEW is reconstructed at an update rate of 40Hz by using the transmitted SEW codebook index. Once the SEW is decoded, it is upsampled to 400Hz

(i.e., having 10 SEWs for each frame) by means of linear interpolation. Each interpolated SEW will have a length of the interpolated pitch.

The REW is reconstructed at an update rate of 400Hz. The decoded SEW and REW are added to render the normalised PW. The complete residual PW is obtained by denormalisation, that is simply a multiplication of the normalised PW and the associated gain term. Then, each PW is time aligned with the previous PW before DFT linear interpolation between them to obtain the complete excitation signal.

The LSF coefficients are decoded using the transmitted codebook indices. To obtain good output speech, as with many coders, decoded LSFs are required to be interpolated before being converted to the reflection coefficients ($-k(i)$). In this coder, LSF interpolation is performed over three continuous frames: Previous-2 Frame, Previous-1 Frame and Present Frame, of which Previous-2 Frame is before Previous-1 Frame. The interpolation scheme is described in Table 3.1.

	Prev-2 Frame	Prev-1 Frame	Pres. Frame
Sub-frame 1	0.4	0.6	0
Sub-frame 2	0.2	0.8	0
Sub-frame 3	1.0	0	0
Sub-frame 4	0	0.8	0.2
Sub-frame 5	0	0.6	0.4

Table 3.1 The LSF Interpolation Scheme of MPW Coders

The output speech signal is produced by synthesising this excitation using a LP synthesis filter, which is an IIR filter with the system function of:

$$H(z) = 1\Big/\left(1 - \sum_{k=1}^{P} a(k)z^{-k}\right) \tag{3.11}$$

such that the synthesis is presented as:

$$s(n) = r(n) + \sum_{k=1}^{P} a(k)s(n-k). \tag{3.12}$$

This filter is also a lattice structured filter (described in Figure 3.4). At low bit rate coding, it is useful to use a postfilter for perceptual quality enhancement of the synthesised speech. As the speech formants are perceptually more important than the formant nulls, the use of a postfilter is to preserve the formant information by keeping the noise in the formant regions as low as possible. Thus, the noise is shaped, and the perceptual quality of the synthesised speech is improved [75-77].

3.2 Pitch Detection

Pitch detection plays an important role in MPW coders as it determines the number of DFT coefficients of the PW, SEW, REW in the DFT domain. The pitch detection technique used in this work is based on that of the following authors: Dubnowski, et.al. [71], Burnett and Holbeche [37], and Gambino and Burnett [72-73]. Briefly, it can be described as follows:

1. The input speech is filtered using a 65 tap low-pass FIR filter, the cut-off frequency of which is 900Hz.

2. A rectangular window of 300 samples centred on the current frame of 200 samples is selected. This, therefore, leads to an overlap of 50 samples in the pitch processing frames.

3. Two segments of 100 samples at the ends of the 300 sample window is processed to find the maximum of both segments. A fixed percentage (80%) of the minimum of these two values is set as the clipping level.

4. Centre clip the section of speech using the clipping level.

5. The expected pitch value, ranging from 20 to 147, is calculated using the autocorrelation function of the centre-clipped signal. The pitch of the speech segment is determined by the maximum of this autocorrelation function.

6. The voiced/unvoived decision is determined by normalising the maximum autocorrelation to the zeroth autocorrelation. The frame is decided to be voiced if the normalised value is larger than 0.28.

These are the basic steps for determining pitch value in each 25ms frame of speech. The pitch value is then converted to a number of samples in each prototype waveform or number of DFT coefficients of PWs.

3.3 Prototype Waveform Extraction and DFT Transform

Prototype waveform extraction can be carried out by several methods. The method chosen for this work uses the linear prediction residual signal and is based on the technique developed by Burnett and Holbeche [37]. In this method, ten PWs are extracted every 20 samples from each 200 sample frame. The pitch ranges from 20 to 147. Thus, when the pitch is 20 there would be up to 10 different PWs and when the pitch is 147 there would be just one PW in the frame. Experiments in this work have shown that when the update rate is less than one PW per frame, the coded speech quality degrades rapidly. This conclusion is also consistent with that reported in [40].

For extraction, a number of PWs around each extraction point are regarded as the candidates. The one with the minimum squared error between its two end points is chosen. To maximise capture of the dynamic nature of speech [40], the number of candidate PWs should be limited such that they are close to the extraction point. In the discrete time domain, at point m, a chosen unaligned prototype, $pw_m(n)$, can be defined as a pitch length segment of the discrete residual signal centered near the discrete time m and at the value n, $0 \le n \le p_m$, (where p_m is the estimated pitch at time m). The extraction of such a residual PW is based on the following formula (adapted from that proposed by Kleijn and Haagen [40]):

$$pw_m(n) = r(m - \frac{p_m}{2} + n + \Delta) \qquad 0 \le n \le p_m \tag{3.13}$$

where $r(n)$ is the LP residual signal and Δ is the searching region.

Since the PW is extracted in the residual domain, this method has advantages over other domains such as the speech domain. In the residual domain pitch pulses are clear and the power between them is low [40].

The speech $s(n)$, as well as the residual speech signal $r(n)$, are periodic and band-limited with bandwidth $Wp_m/2\pi F_s$ (radians^{-1}), where the bandwidth of the speech is W (in Hz), and the sampling frequency of the speech is F_s (in Hz). Thus, the extracted residual PWs can be described by a finite discrete Fourier series:

$$pw_m(n) = \sum_{k=0}^{P_m-1}[A_m(k)\cos(\frac{2\pi kn}{p_m}) + B_m(k)\sin(\frac{2\pi kn}{p_m})] \tag{3.14}$$

or in polar notation

$$pw_m(n) = \frac{1}{p_m}\sum_{k=0}^{p_m-1} PW_m(k)\exp(j2\pi kn/p_m) \tag{3.15}$$

where $PW_m(k) = \sqrt{|A_m(k)|^2 + |B_m(k)|^2}$. The coefficients $PW_m(k)$ for the prototype waveform in the DFT domain can be derived from (3.15) as:

$$PW_m(k) = \sum_{n=0}^{P_m-1} pw_m(n) \exp(-j2\pi kn/p_m). \quad (3.16)$$

The real and imaginary values of the DFT coefficients $PW_m(k)$ are derived from Equation (3.16) as:

$$\mathrm{Re}[PW_m(k)] = \sum_{n=0}^{p_m-1} pw_m(n) \cos(2\pi kn / p_m) \quad (3.17a)$$

$$\mathrm{Im}[PW_m(k)] = -\sum_{n=0}^{p_m-1} pw_m(n) \sin(2\pi kn / p_m). \quad (3.17b)$$

3.4 Alignment of Prototype Waveforms

In prototype waveform coding, time alignment of the prototype waveforms is required for smooth reconstruction of the residual signal. This section describes the method used to correct the currently extracted PW to be time aligned with the previous PW. The method described here is an adaptation of the technique proposed by Kleijn [36] and developed by Burnett and Bradley [41]. The procedure can be described as follows:

Once a residual prototype $pw_m(n)$ is extracted, it is transformed into DFT series $PW_m(k)$ of length p_m. For time alignment with the previous prototype, $pw_{m-1}(n)$, whose Fourier coefficients are $PW_{m-1}(k)$ of length p_{m-1} the cross-correlation between them must be maximised.

To align two PWs, they must be described by an equal number of coefficients in the DFT domain. The problem of unequal length PWs can easily be solved by means of zero-padding. If the two PWs are of unequal length, the PW with

the lower number of harmonics is padded with harmonics of zero amplitude. When p_m and p_{m-1} are related by a factor of 2 or 3, the PW with the lower number of harmonics is repeated such that both PWs have the same length [36]. This adjustment technique plays an important role in dealing with pitch doubling/halving in speech since it avoids unnatural interpolation.

After adjustments, two unaligned prototype waveforms $PW'_{m-1}(k)$ and $PW'_m(k)$ of adjusted length p are produced. In the time domain the alignment can be considered as a rotation, while in the DFT domain it is equivalent to a phase shift of $PW'_m(k)$ [36]. In other words, the DFT coefficients of the present prototype waveform, $PW'_m(k)$, need to be time-shifted by a time interval τ such that cross correlation with the DFT series of the previous prototype waveform, $PW'_{m-1}(k)$, is maximised. This time shift is calculated by finding the maximum:

$$\tau = \underset{\tau'}{argmax} \sum_{k=0}^{p} \{\mathrm{Re}[PW'_m(k)PW'^{*}_{m-1}(k)]\cos(2\pi k\tau') + \mathrm{Im}[PW'_m(k)PW'^{*}_{m-1}(k)]\sin(2\pi k\tau')\} \qquad (3.18)$$

$$for\ \tau' = 0.001,...,1.$$

It is convenient to normalise τ' to the pitch period, thus simplifying the calculation of Equation (3.18). Once the time shift τ, which maximises the DFT cross correlation of Equation (3.18), is found the aligned present prototype waveform, $\tilde{PW}'_m(k)$, is calculated by applying τ to phase shift the unaligned present prototype waveform, $PW'_m(k)$:

$$\tilde{PW}'_m(k) = PW'_m(k)\exp(j2\pi k\tau) \qquad for\ k = 0, 1,... \tau. \qquad (3.19)$$

It should be noted that this time alignment technique prevents impulsive distortions in interpolated and synthesised speech. On the other hand it changes the position of PW. Thus, the information about the absolute position of the PW is removed and is not transmitted. Hence, the synthesised speech

waveform is not synchronous with the input speech. The perceived speech quality is, however, invariant with the phase shift of the output speech [40].

3.5 Interpolation of Prototype Waveforms

This section describes the technique employed to smoothly interpolate the prototype waveforms in the DFT domain for reconstruction of the speech. At the encoder, the time aligned PWs are quantised by a certain quantisation scheme. At the decoder, ten PWs are reconstructed each frame. These are then linearly interpolated to obtain the complete excitation signal which, when filtered by a synthesis LP filter, result in speech being reproduced. The continuous interpolation between prototype waveforms described here is based on the algorithm proposed by Kleijn [34] and used by Burnett and Bradley [41]. It can be presented as follows:

Given two time aligned PWs in the DFT domain: $PW_{m-1}(k)$ and $PW_m(k)$ with lengths (pitch periods) p_{m-1} and p_m respectively, the pitch period of the successively interpolated PW at a given interpolation point, i, between PWs can be interpolated as:

$$p_i = (1-\alpha_i)p_{m-1} + \alpha_i p_m$$
$$for\ m-1 \le i \le m;\ 0 \le \alpha_i \le 1 \tag{3.20}$$

where α_i is a monotonically increasing interpolation function [34] in the time between the two PWs. α_i is regarded as the coefficient describing the contribution level of the present prototype waveform $PW_m(k)$ to the interpolated prototype waveform at time i in terms of harmonic magnitude and PW's length p_i. In the same manner, $(1-\alpha_i)$ describes the contribution level taken from the previous prototype waveform $PW_{m-1}(k)$ [41].

The smooth DFT interpolation between the previous prototype waveform $PW_{m-1}(k)$ and the aligned m^{th} prototype waveform $PW_m(k)$ over the interpolation interval L will result in the continuously interpolated excitation $e_m(i)$ in the time domain. Based on the interpolated pitch p_i the interpolation algorithm is found to be:

$$e_m(i) = \sum_{k=0}^{p_i-1} \{\mathrm{Re}[\alpha_i PW_m(k) + (1-\alpha_i) PW_{m-1}(k)]\cos(\frac{2\pi ki}{p_i}) + \mathrm{Im}[\alpha_i PW_m(k) + (1-\alpha_i) PW_{m-1}(k)]\sin(\frac{2\pi ki}{p_i})\}$$

for i=0,...,L-1. (3.21)

The interpolation interval L was chosen to be one tenth of the frame length (20 samples). This interval is identical to the shortest pitch of 20. Figure 3.5 shows an example of the processing of a number of voice speech frames by the MPW coder without quantisation. It can be seen that the interpolated excitation and the output speech are close to the original excitation and the input speech respectively, however, as expected, the output speech and the interpolated excitation are not in synchronisation with the input speech and the residual signal due to time alignment effects.

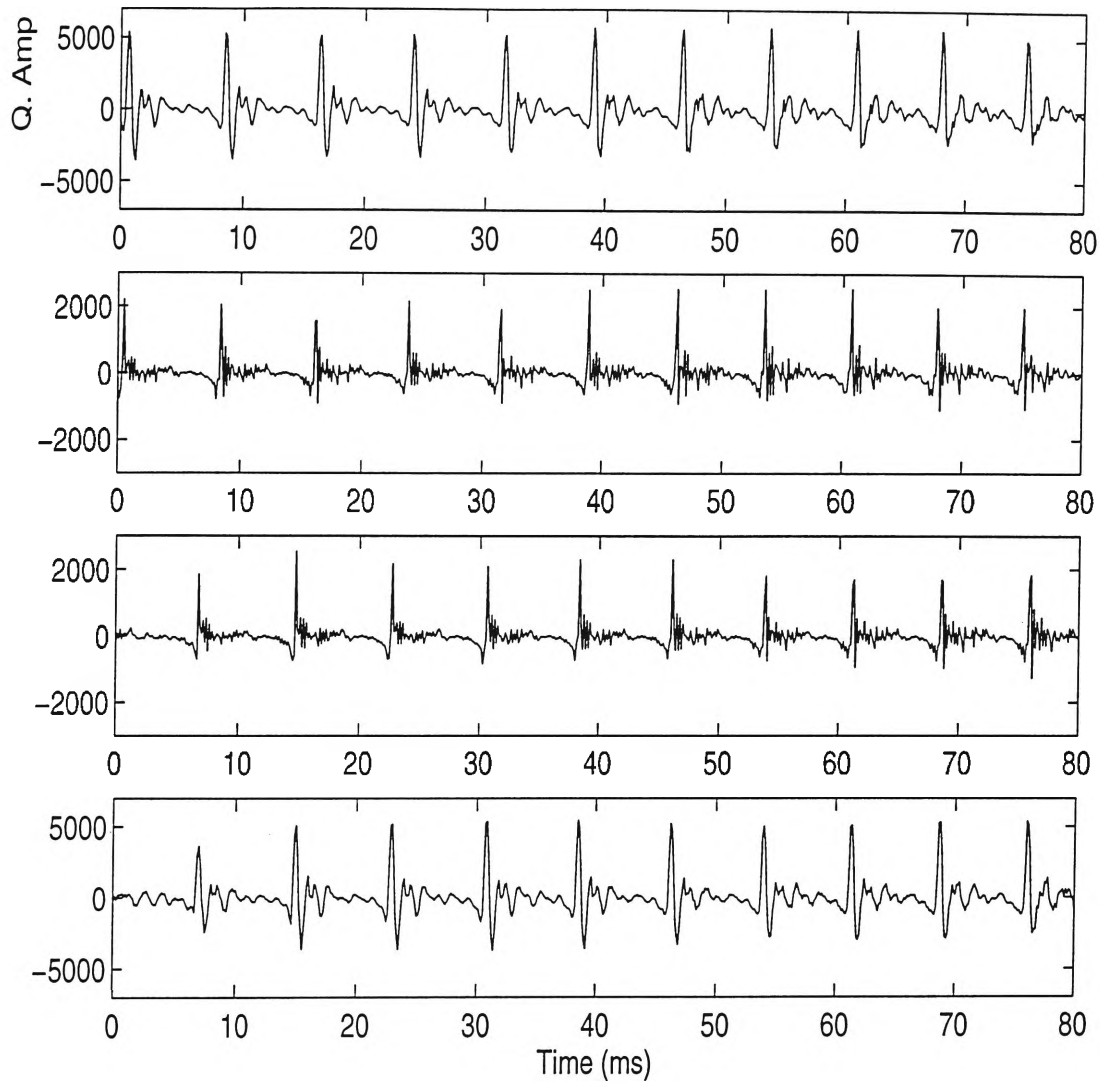


Figure 3.5 Example of a Voiced Speech Frame Processed by the MPW Coder Without Quantisation (From the top to the bottom: input speech, residual, reconstructed excitation, and reconstructed speech)

3.6 Decomposition of Prototype Waveforms

In MPW coders, the speech signal is represented as a sequence of prototype waveforms. Without quantisation, perfect quality speech can be reconstructed from these PWs. The quality of this reconstruction does not depend on the initial phase of the PWs. For efficient quantisation, it is required to decompose the prototype waveform into components with distinct properties. This allows the separate components to be quantised independently according to their

characteristics. Bit allocation for the quantisation of each component is dependent on its requirements and hence high quality speech at low bit rates is obtainable. For such a decomposition to be effective, it is necessary to consider the perceptual information of the speech signal and hence the prototype waveform. During unvoiced speech, the phase spectrum of the PW changes rapidly, while in voiced speech the PW evolves slowly. Generally, the speech signal is a combination of voiced signal (quasi-periodic signal) and unvoiced signal (noise-like signal). Prototype waveform coding has been found to perform best if the decomposition is performed during both voiced and unvoiced speech [40-45].

Kleijn and Haagen [42] described a decomposition in which a residual prototype waveform (PW) is decomposed into a slowly evolving waveform (SEW) and a rapidly evolving waveform (REW). The SEW can be regarded as the underlying pulse shape of the PW while the REW is representative of the noisy components in the PW. In practice, the decomposition can be performed by filtering the DFT coefficients of the PWs using a low-pass filter (for SEW) and a high-pass filter (for REW) with a suitable cut-off frequency of 20Hz. Such a SEW/REW paradigm is one of the accepted mechanisms for decomposition of the PW. The SEW can be quantised at low update rates, while the REW phase is noise-like and hence need not be transmitted in detail. The REW magnitude evolves rapidly, however, it contributes to the PW shape, thus it should be quantised with a certain update rate according to available bit rates.

The separate REW and SEW components must sum to the entire PW:

$$PW_m(k) = REW_m(k) + SEW_m(k) \tag{3.22}$$

where $PW_m(k), SEW_m(k)$ and $REW_m(k)$ are the DFT series of the PW, SEW and REW respectively.

An alternative SEW, defined by Burnett and Bradley [41], is formed as a 'mean' PW of ten residual PWs extracted in each 25ms frame. The REW is then the noise-like remainder of the extracted PWs. Expression for deriving the DFT coefficients of the SEW from the DFT coefficients of the extracted PWs can be presented as:

$$SEW(k) = \frac{1}{10}\sum_{m=1}^{10} PW_m(k) \qquad for\ k = 0,\ldots,p_m - 1 \tag{3.23}$$

where p_m is the length (number of DFT coefficients) of $PW_m(k)$. As the SEW is formed as a mean PW, the DFT coefficients of the REWs are calculated as:

$$\begin{aligned} &REW_m(k) = PW_m(k) - SEW(k) \\ &for\ m = 1,\ldots,10;\ \ k = 0,\ldots,p_m - 1. \end{aligned} \tag{3.24}$$

This definition of SEW and REW has been reported to be advantageous when pitch detection is not reliable and doubling and halving cannot be detected [41]. The simplicity of this definition is attractive for low bit rate coding.

Both definitions of the SEW/REW lead to the separation of the PW into two components, an underlying pitch pulse and a noise-like waveform [41]. However, the latter definition can significantly reduce the complexity of the decomposition process. In the latter, the SEW can be obtained by a simple computation and the REWs are the differences between the SEW and the extracted PWs, whereas in the former definition the SEW and REW are filtered using low-pass and high-pass filters.

In MPW coders, ten residual PWs are extracted, and then decomposed into SEWs and REWs each frame. Since the SEW is a slowly evolving component, for low bit rate coding, the ten SEWs are required to be downsampled to an

update rate of 40Hz. In terms of complexity, the definition of SEW as a 'mean' prototype waveform should thus be chosen for this work.

In this work, the PWs are extracted at a rate of 400Hz, and then transformed to the DFT domain. After time alignment, the gain term or power of each PW is calculated as:

$$G_m = \frac{1}{p_m} \sum_{k=0}^{p_m-1} PW_m(k) PW_m^*(k) \qquad for\ m = 1,...,10. \tag{3.25}$$

The time aligned PWs are normalised according to their gain terms. If $PW_{m(norm)}(k)$ is defined as the DFT coefficients of the normalised PWs, the formula for normalisation is:

$$PW_{m(norm)}(k) = \frac{PW_m(k)}{G_m}$$
$$for\ k = 0,..., p_m - 1;\ \ m = 1,...,10. \tag{3.26}$$

For each frame, the SEW is formed as the average PW of the ten normalised PWs, can thus be found using Equation (3.23), and the REW is calculated using Equation (3.24), but in both equations, $PW_{m(norm)}(k)$ is used instead of $PW_m(k)$. As an example, Figure 3.6 describes the decomposition of a PW into a SEW (the 'mean' PW) and REW (the noise remainder of the extracted PW) in three dimensional space. The SEW, as expected, looks smoother than the PW, and the REW is noise-like.

A typical characteristic of prototype waveform coding is that speech or residual and, hence, a series of PWs, can be represented as a two-dimensional signal. The decomposition of the PW into a SEW and REW can also be considered as a type of sub-band coding, but rather than an ordinary, one-dimensional sub-band coding, it is a two-dimensional sub-band coding. The low band forms the SEW, and the high band forms the REW with a cut-off

frequency of about 20Hz. However, from the viewpoint of coding each individual PW, the REW/SEW paradigm is no different from ordinary sub-band coding. This concept suggests that quantisation of prototype waveforms can be performed using sub-band coding techniques. However, this concept has arisen during this work, and has not been investigated yet.

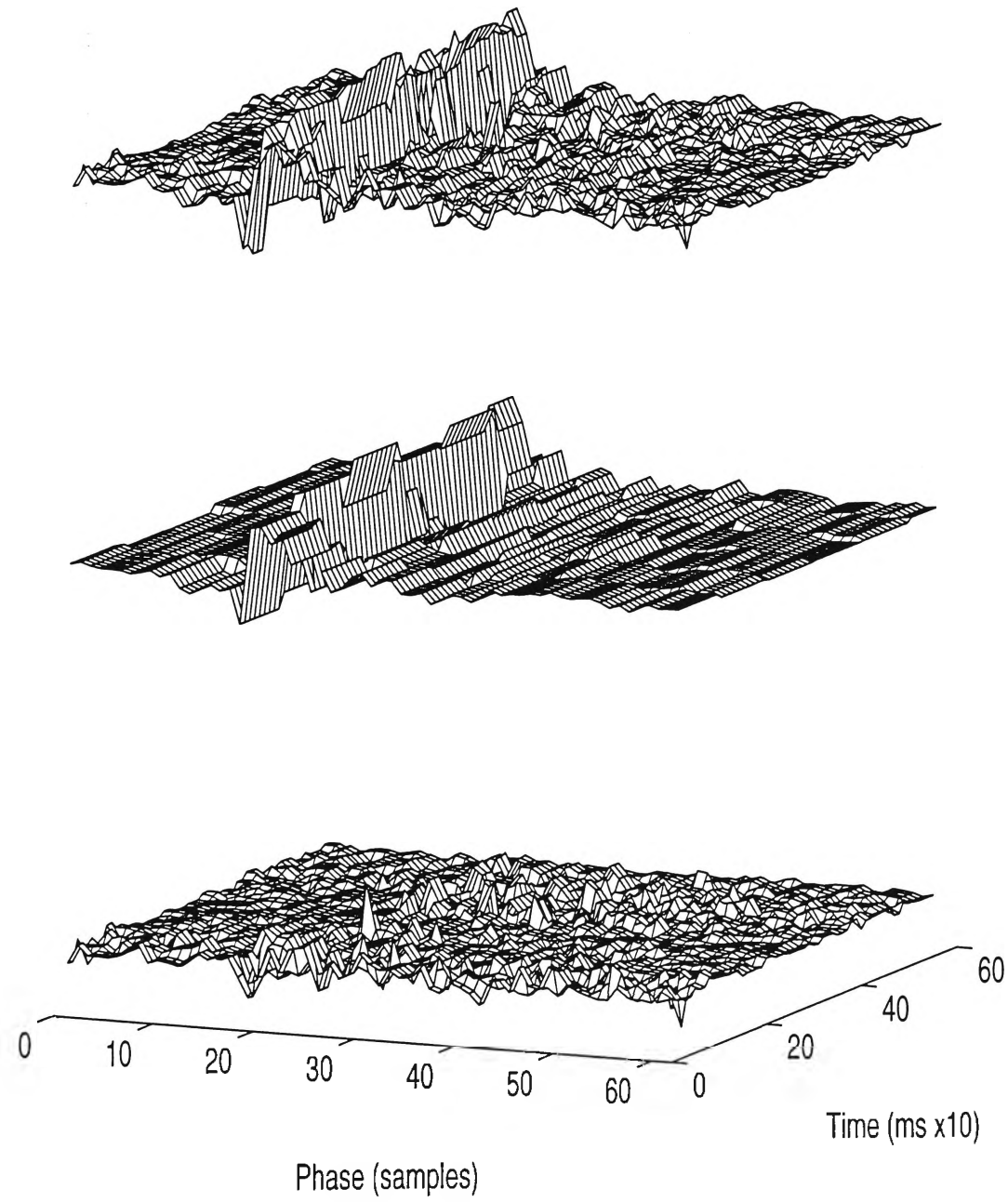


Figure 3.6 Decomposition of Residual PWs into SEWs and REWs using the Definition of the SEW as a 'mean' PW (From the top to the bottom: PWs, SEWs and REWs)

3.7 Summary

The principles of Prototype Waveform Coding have been discussed. In MPW coders, the PWs are extracted from residual the signal at a rate of 400Hz in both voiced and unvoiced speech. The extracted PWs are DFT transformed, time aligned, and then normalised. For low bit rate quantisation, it is essential that the PWs be decomposed into a number of distinct components, the quantisation requirements of which are dependent on their characteristics. The excitation signal can be reconstructed by continuous interpolation between the reconstructed PWs.

Prototype waveform extraction can be performed in various ways. The technique used in this work allows a PW to be extracted correctly. Time alignment enables the extracted PW to be time aligned with the previous PW. Although the length of PWs varies with the pitch period of speech, it does not cause a major problem. For the PWs to be the same length for time alignment, it is possible to zero-harmonic pad to the shortest PW length. When the PWs are related to each other by a factor of 2 or 3, the shortest PW can be repeated by 2 or 3 such that they have equal length [36].

The decomposition of prototype waveforms can be performed in various ways. One of the accepted decomposition paradigms is to decompose the PW into a slowly evolving waveform (SEW) and a rapidly evolving waveform (REW) using low-pass and high-pass filters with a cut-off frequency of 20Hz [42]. Alternatively, the SEW can be formed as the average PW of the extracted PWs each frame, and the REW is the remainder of the extracted PWs [41]. This definition has been chosen for the work described in this thesis because of its low decomposition complexity.

At the receiver, the excitation signal can be reconstructed from the PWs by smooth linear interpolation. Operating in the DFT domain has a drawback in

computational cost, however, good performance of interpolation has been obtained [40-45].

Chapter 4: Multi-Prototype Waveform Open Loop Coding

In Chapter 3, the architecture and principles of operation of a Multi-Prototype Waveform (MPW) coding system were described. The main issue in such a system is how to quantise the prototype waveforms for low bit rate coders. This chapter proposes two quantisation schemes in an open loop architecture for prototype waveforms, called the Unity Magnitude Quantisation and the Errored Magnitude Quantisation. At bit rates of 2.4kb/s, the Open Loop Quantisation based MPW coders: the Unity Magnitude Coder and the Errored Magnitude Coder have been shown to be capable of producing good quality communication speech. The performance of these coders is evaluated using Mean Opinion Scores and is compared with the US 1016 Federal Standard CELP-4.8kb/s. A key to achieving good quality speech is the construction of effective codebooks for SEW/REW. The 8 bit codebook for SEW consists of both SEW magnitude and phase spectra for each vector, while the 5 bit REW/*Error* codebook consists of magnitude only for each vector. An essential part of MPW coders is quantisation of the LPC spectrum. Thus, quantisation of Line Spectral Frequencies is also discussed in this chapter.

4.1 Unity Magnitude Quantisation

This section describes the first quantisation scheme for prototype waveforms, Unity Magnitude Quantisation, and the Unity Magnitude Coder which was developed on the basis of this quantisation scheme.

4.1.1 Prototype Waveform Quantisation

As discussed in the previous chapter, for effective quantisation at low bit rates, the extracted PW is required to be decomposed into a SEW and a REW. In this scheme, the definition of the SEW as a 'mean' PW is used. Quantisation of the SEW and REW is now discussed.

4.1.1.1 Quantisation of SEW

In each 25ms frame, the SEW is formed as the average PW of the ten normalised PWs using the following equation:

$$SEW(k) = \frac{1}{10}\sum_{m=1}^{10} PW_{m(norm)}(k) \qquad for\ k = 0,\ldots,p_m - 1. \qquad (4.1)$$

Since vector quantisation always achieves better performance over scalar quantisation [53], in this work, quantisation of the SEW is based on vector quantisation techniques. According to Chang, et.al. [46], VQ of the speech signal performed in the DFT domain has two advantages. These advantages can also be extended to the prototype waveform (hence SEW/REW). Firstly, each sample in the DFT domain is a combination of all samples of the speech waveforms in the time domain. Thus, quantisation of the samples in the DFT domain can provide better performance than quantisation of the samples in the time domain. Secondly, VQ on the DFT transformed speech waveforms provides distinctly better subjective quality than VQ on the speech waveform in other representations.

It should be noted that a DFT coefficient can be represented either by a pair of real/imaginary coefficients or by a pair of magnitude/phase coefficients. According to these authors [46], for speech coding, VQ on the magnitude and phase coefficients can yield better subjective quality than VQ on the real and imaginary coefficients with the same chosen bit rate.

The amplitude and phase characteristics of the SEW vary with the nature of voicing in the speech, thus, the coded speech quality mainly depends on the SEW [74]. Since there is no priority between the SEW magnitude and SEW phase in the reconstruction of the underlying pulse-shape of the PW, it is necessary to quantise both the magnitude and phase spectra of the SEW.

In this work, the quantisation of the SEW using an 8 bit codebook (256 vectors) for both SEW magnitude and phase spectra has been found to be the best solution for MPW coders operating at 2.4kb/s. However, at higher rates, quantisation of the SEW magnitude and phase separately using two different codebooks is preferable [78]. Normally, as the codebook is larger, its performance is better [53]. The size of the SEW codebook is chosen based on experiments. However, investigations in this work show that the performance of the 9 bit SEW codebook is close to that of the 8 bit SEW codebook. Degradation of performance is significant when the codebook size is reduced to 7 bits.

During unvoiced frames, as the noisy REW dominates, the SEW is flat and its power is much smaller than that in voiced frames. Such a SEW must be different from the SEW used during voiced frames. This needs to be considered during the design of the SEW codebook; as the codebook needs to effectively quantise unvoiced speech. To allow this, the codebook has two sections; the first one consists of 40 SEW vectors for unvoiced speech and the second, 216 SEW vectors for voiced speech.

Since the DFT coefficient series of the SEW are symmetrical, it is possible to quantise only half of the DFT series of a SEW. Before searching the SEW codebook, the extracted SEW must be aligned with a standard vector since the codebook vectors were already aligned with this vector (see Section 4.3.1). For simplification, a squared error distortion measure is used for the codebook search. The coder chooses the codebook vector whose mean squared error $d(SEW,\hat{SEW})$ is minimum. Such a distortion measure is presented in the following equation.

$$d(SEW,\hat{SEW}) \equiv \sum_{k=0}^{p_m/2}\{[SEWmag(k)-\hat{SEWmag}(k)]^2 + [SEWpha(k)-\hat{SEWpha}(k)]^2\}. \tag{4.2}$$

It is possible, however, to use a weighted squared error distortion measure for improved performance. The weighted squared error distortion measure is:

$$d(SEW,\hat{SEW}) \equiv \sum_{k=0}^{p_m/2} w(k)\{[SEWmag(k)-\hat{SEWmag}(k)]^2 + [SEWpha(k)-\hat{SEWpha}(k)]^2\} \tag{4.3}$$

where SEW, $\hat{SEW}$ are the input vector and the codebook vector respectively. The weighting function $w(k)$ is dependent on the $SEWmag(k)$ and $SEWpha(k)$.

In terms of reproducing the characteristic waveform, however, the SEW spectral magnitude is more important than the SEW spectral phase due to the human ear being more sensitive to the magnitude coefficients than the phase coefficients [46]. Thus, the codebook search could be performed on the SEW spectral magnitude alone, while the SEW phase spectra is followed by the selection of the SEW magnitude spectra. In this case the distortion measure is simplified as:

$$d(SEW,\hat{SEW}) \equiv \sum_{k=0}^{p_m/2}[SEWmag(k)-\hat{SEWmag}(k)]^2 . \tag{4.4}$$

4.1.1.2 Quantisation of REW

The REW is a noise-like component which is rapidly evolving and has a quantisation requirement lower than that of the SEW. The coded speech quality is mainly dependent on the SEW, while the REW determines the naturalness and the dynamics of the speech [74]. The REW phase spectrum changes rapidly, and can be considered as noise, and at the decoder, can be derived from Gaussian noise [79]. The REW magnitude spectrum, however, contributes to the overall structure of the prototype waveform, and should be transmitted as accurately as possible according to the number of bits available.

As the DFT series of the extracted PW is normalised, the average magnitude of the normalised PW is equal to unity. This suggests that the magnitude of the normalised PW can be approximated to be unity. For coding at bit rates as low as 2.4kb/s, the magnitude spectrum of the REW can be derived on the basis of this approximation:

$$REWmag_m(k) = 1.0 - SEWmag_m(k)$$
$$for\ k = 0,\ldots,p_m;\ \ m = 1,\ldots,10. \tag{4.5}$$

Thus, in this quantisation scheme, only the SEW is quantised, the REW is not quantised. At the decoder, the decoded SEW is interpolated from an update rate of 40Hz to 400Hz (i.e., ten SEWs per frame). The REW magnitudes can be effectively reconstructed with an update rate of 400Hz based on the ten reconstructed SEW magnitudes by using Equation (4.5).

4.1.2 Unity Magnitude Coder

This section discusses the operation of the Unity Magnitude Coder; the architecture of which is described in Figure 4.1.

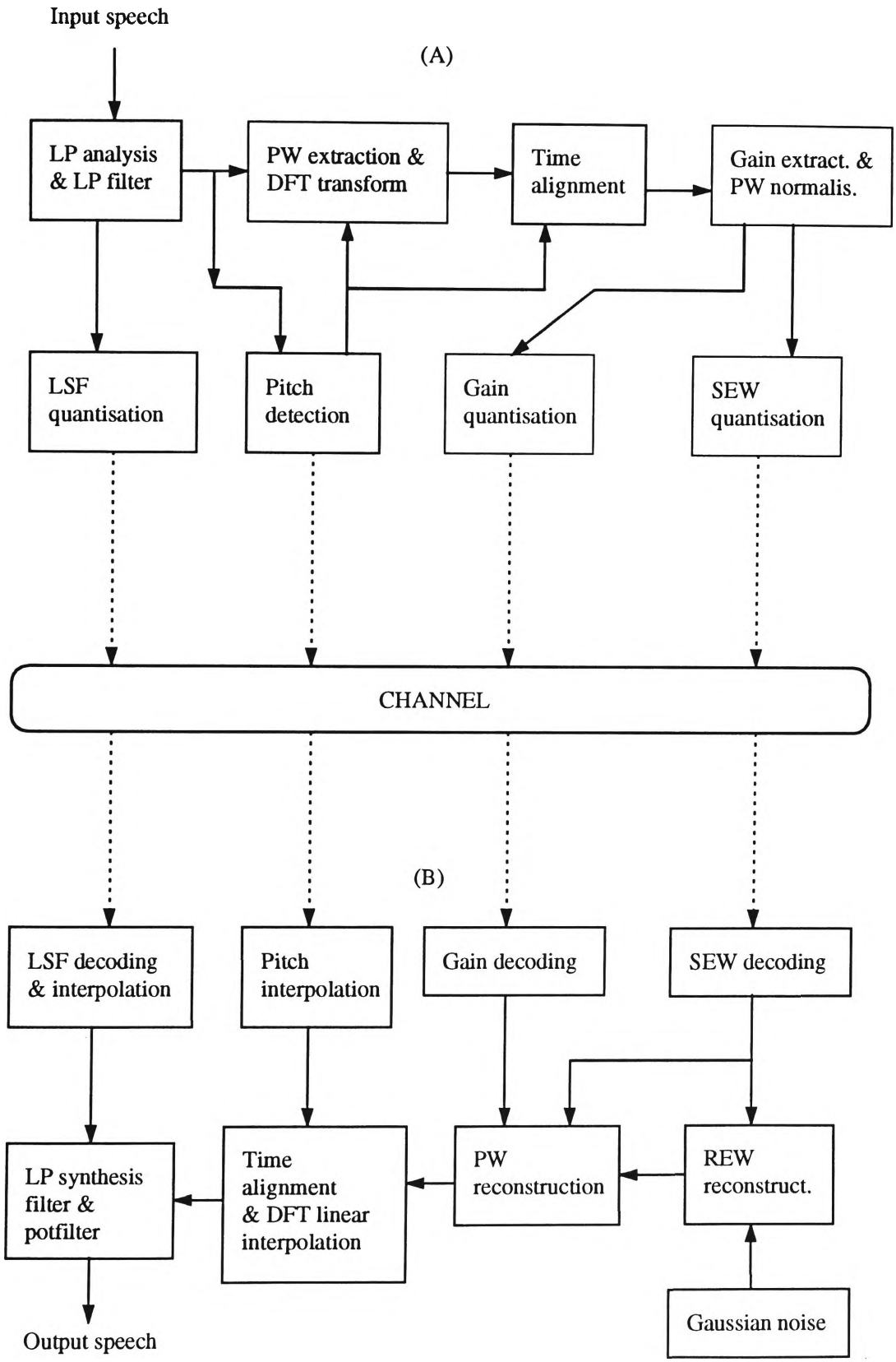


Figure 4.1 Unity Magnitude Coder Architecture: (A) Encoder, (B) Decoder

The Encoder processes the speech on a 25ms frame basis. For every frame, 10^{th} order LPC coefficients are extracted. The speech is analysed by a lattice LP analysis filter to obtain the LP residual signal. The PWs are extracted from the residual signal at a rate of 400Hz (i.e., 10 PWs per frame), DFT transformed, time aligned and then normalised. The gains are extracted, the logarithms taken and then downsampled to 120Hz (i.e., three gain terms per frame), and then quantised using a 5 bit SQ. For quantisation, the normalised PW is decomposed into a SEW and a REW. The SEW is formed as the average PW of the ten extracted PWs. As a result, it is downsampled to 40Hz (i.e., one SEW per frame). The SEW is then quantised using the 8 bit SEW codebook. The LSFs are quantised using the 30 bit/frame split-VQ.

At the Decoder, decoded LSFs are interpolated before being converted to reflection coefficients. The SEW is decoded using its codebook index, and then upsampled to 400Hz by means of linear interpolation according to the interpolated pitches. The REW magnitudes are reconstructed at a rate of 400Hz according to SEW behaviours and the unity approximation using Equation (4.5). The REW phases are derived from a Gaussian noise source. The sum of the SEW and REW renders the normalised PW. The complete PW is obtained by multiplying the normalised PW with the gain terms. The complete excitation signal is obtained by continuous interpolation in the DFT domain between the time aligned PWs. Finally, the excitation is synthesised using a lattice LP synthesis filter. Perceptual quality of the coded speech can be enhanced by using a postfilter.

The bit allocation for this coder is given in Table 4.1. In this plan, the total bits is 60 per 25ms frame. This is equivalent to a total bit rate of 2.4kb/s.

Parameter	Number of bits	Update rate
LSF	30	40 Hz
pitch	7	40 Hz
gain	5	120 Hz
SEW	8	40 Hz

Table 4.1 Bit Allocation for the Unity Magnitude Coder.

4.2 Errored Magnitude Quantisation

This section discusses the second prototype waveform quantisation scheme: Errored Magnitude Quantisation, and describes the Errored Magnitude Coder.

4.2.1 Prototype Waveform Quantisation

In the Unity Magnitude Quantisation scheme, the REW magnitude is not transmitted, and is thus recovered on the basis of the decoded SEW magnitudes and the assumption that the magnitude of the normalised PW is unity. The Unity Magnitude Coder codes the gain terms using a 5 bit SQ at an update rate of 120Hz. As the gain term changes slowly with time, it can be transmitted at a rate of 80Hz rather than 120Hz without any significant speech quality degradation. Informal listening tests were performed on 14 TIMIT speech files. During the tests, the listeners did not recognise any differences between the speech coded using the gain term with an update rate of 120Hz and that coded using the gain term with an update rate of 80Hz. This scheme investigates the use of the redundant bits in transmission of the gain terms to quantise the REW magnitude. This quantisation scheme is considered an empirical method aimed at making overall perceptual quality improvements to the Unity Magnitude Quantisation scheme.

As the REW is regarded as the difference between the normalised PW and the SEW, the REW magnitude can be formed as:

$$REWmag_m(k) = PWmag_m(k) - SEWmag_m(k)$$
$$for\ k = 0,\ldots,p_m - 1;\ \ m = 1,\ldots,10. \tag{4.6}$$

This equation can also be written as:

$$REWmag_m(k) = [PWmag_m(k) - 1] + [1 - SEWmag_m(k)]$$
$$for\ k = 0,\ldots,p_m - 1;\ \ m = 1,\ldots,10. \tag{4.7}$$

Substituting:

$$Error_m(k) = PWmag_m(k) - 1.0$$
$$for\ k = 0,\ldots,p_m - 1;\ m = 1,\ldots,10 \tag{4.8}$$

into Equation (3.36), the REW magnitude can be presented as:

$$REWmag_m(k) = [1 + Error_m(k)] - SEWmag_m(k)$$
$$for\ k = 0,\ldots,p - 1;\ \ m = 1,\ldots,10. \tag{4.9}$$

From Equation (4.9), it can be seen that the REW magnitude now depends on the SEW magnitude and the *Error*. As the *Error* is defined as the magnitude of the normalised PW after the removal of the mean, it is representative of the evolution (the dynamics) of the PW magnitude. In terms of subjective quality, informal listening tests in this work have shown that VQ on the *Error* provides better performance over VQ directly, either on the prototype magnitude or on the REW magnitude. During the tests, the listeners preferred speech coded using the *Error* quantisation to speech coded using the quantisation of the magnitude of the REW/PW.

In this scheme, the *Error*, $Error_m(k)$, is extracted at a rate of 400Hz using Equation (4.8) then downsampled to a certain rate dependent on the bit allocation scheme of the coder. In this case, it is downsampled to 40Hz and then quantised using a 5 bit VQ. At the decoder, the REW magnitude is recovered using this *Error* according to Equation (4.9).

4.2.2 Errored Magnitude Coder

This section discusses the operation of the Errored Magnitude Coder; the architecture of which is described in Figure 4.2. Basically, this coder is similar to the Unity Magnitude Coder except for the quantisation of the *Error*.

For each 25 ms frame, the Encoder extracts PWs from the residual at a rate of 400Hz. After DFT transformation, time alignment and normalisation of the PWs, the SEW is extracted and the *Error*s are formed as the difference between the magnitude of the normalised PW and unity using Equation (4.8). The logarithms of the gains are taken and quantised using the 5 bit SQ at an update rate of 80Hz. The SEW is quantised using the 8 bit SEW codebook and the *Error* is quantised using the 5 bit *Error* VQ at an update rate of 40Hz.

At the Decoder, the SEW is decoded using its codebook index, and then upsampled to 400Hz according to the interpolated pitches. The REW magnitudes are reconstructed at 400Hz using Equation (4.9), and the REW phases are derived from a Gaussian noise source.

The bit allocation for this coder is given in Table 4.2. This bit allocation plan is of 60 bits per 25ms frame and equivalent to a total bit rate of 2.4kb/s.

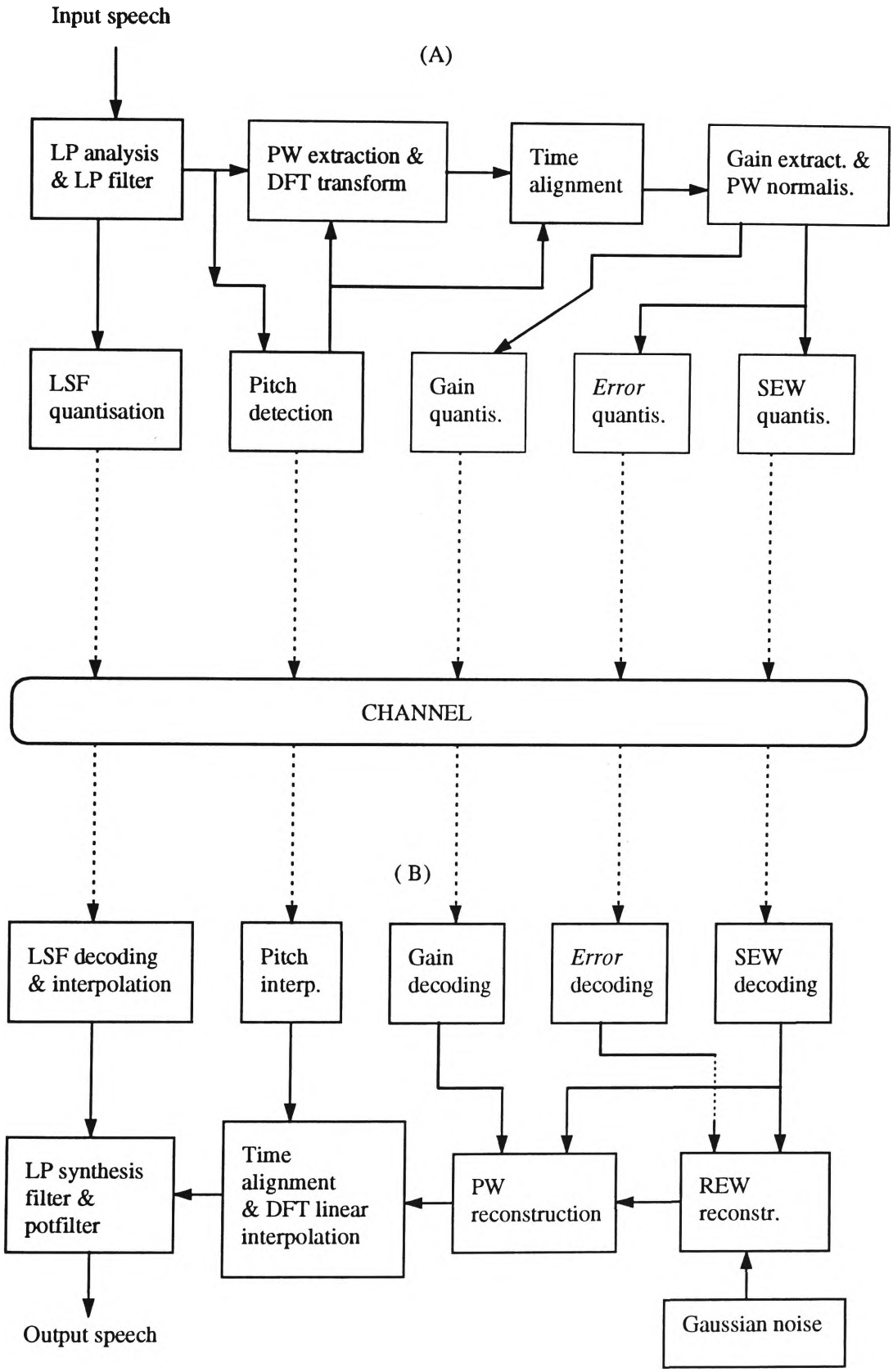


Figure 4.2 Error Magnitude Coder Architecture: (A) Encoder, (B) Decoder

Parameter	Number of bits	Update rate
LSF	30	40 Hz
pitch	7	40 Hz
gain	5	80 Hz
Error	5	40 Hz
SEW	8	40 Hz

Table 4.2 Bit Allocation for the Errored Magnitude Coder

4.3 Codebook Solution and Codebook Design

For quantisation of the SEW, REW/*Error*, and LSFs, in this work, vector quantisation is utilised. The codebook design method is based on the LBG algorithm wherein an input set of training data is used to determine the codebook vector such that the expected distortion is minimised. The LBG algorithm is used for designing vector quantisers with a general distortion measure on a long training sequence of data. There is no theoretical optimum for convergence properties of the codebook for both length of the training data and the number of iterations of the algorithm [51]. Thus, in this work, the sets of training data and the number of iterations of the training procedure are selected such that the aim of obtaining good codebooks can be achieved. The key to achieving suitable codebooks for quantisation of the SEW and REW is a codebook solution. This solution is thus discussed initially.

4.3.1 Codebook Solution

The SEW magnitude and SEW phase spectra can be derived from the DFT coefficients of the SEW as:

$$SEWmag(k) = \sqrt{SEW(k).SEW^{*}(k)} \quad for\ k = 0,..., p_m - 1 \qquad (4.10)$$

$$SEWpha(k) = arctg(SEW(k)) \quad for\ k = 0,..., p_m - 1. \qquad (4.11)$$

The *tan(x)* function is a periodic function of π. Its fundamental period is from $-\pi/2$ to $\pi/2$. The $tan^{-1}(y)$ or *arctan(y)* always returns values falling in the range from $-\pi/2$ to $\pi/2$. Thus, the modulo-2π property does not prevent the inclusion of both the SEW phase spectrum and the SEW magnitude spectrum in each codebook vector.

The problem with SEW quantisation is that the PW length, and thus the SEW length vary with pitch from 20 to 147. For codebook training, a simple solution is used by choosing a standard length. Each training vector is zero-padded such that its length is equal to the standard length. As the DFT series of the prototype waveform is symmetrical and the maximum pitch value is 147, the SEW codebook is designed so that each codebook vector has a chosen standard length of 148. This consists of two sections: the first section contains 74 DFT coefficients for the SEW magnitude spectra and the second, 74 DFT coefficients for the SEW phase spectra. The $SEW(k)$ is extracted from the DFT series of the normalised prototype waveform. To enhance the codebook performance, each extracted *SEW(k)* is aligned with a standard vector. This standard vector was simply chosen such that the first coefficient is unity and the remaining 147 coefficients are zero (vector length is 148). This alignment is aimed at guaranteeing an overall phase response match between the input vectors and the codebook vectors. This allows the training process to be performed in a meaningful manner. The time aligned *SEW(k)* is then decomposed into a series of magnitude spectra, $SEWmag(k)$, and phase spectra, $SEWpha(k)$. The first half of each series is zero-padded to a length of 74 and represents half of the training vector. The process for preparing SEW training vectors for training the SEW codebook is summarised in Figure 4.3.

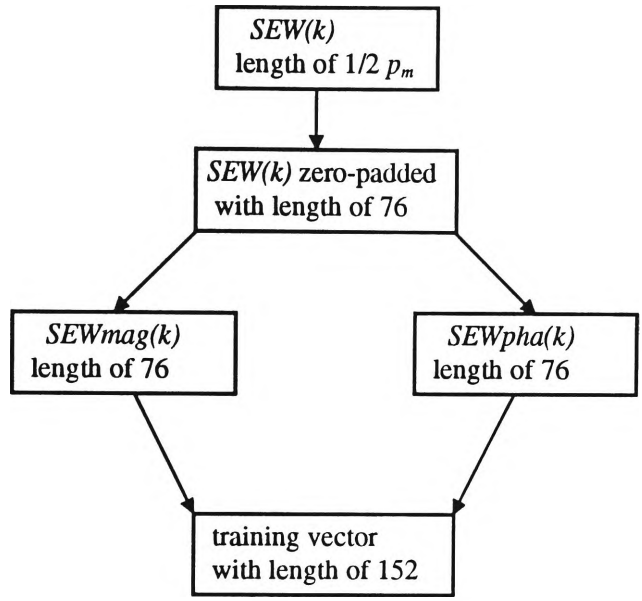


Figure 4.3 The Process for Preparing for the SEW Training Vectors

Unlike SEW quantisation, the REW/*Error* is quantised only on the basis of magnitude spectra using a 5 bit codebook. The *Error* codebook vectors are thus fixed to a standard length of 74. Each codebook vector consists of only the magnitude coefficients.

4.3.2 Distortion Measures

The distortion caused by reproducing an input vector $\mathbf{x}$ by a reproduction vector $\hat{\mathbf{x}}$ is given by the distortion measure $d(\mathbf{x}, \hat{\mathbf{x}})$. There are many distortion measures proposed in the literature. However, for reasons of mathematical convenience the most commonly used is the squared error distortion measure [51]. This measure is also employed in this work for quantisation of the SEW and the REW/*Error*:

$$d(SEW, \hat{SEW}) \cong \sum_{k=0}^{73} \{[SEWmag(k) - \hat{SEWmag}(k)]^2 + [SEWpha(k) - \hat{SEWpha}(k)]^2\} \quad (4.12)$$

for *Error* the formula is:

$$d(Error, \hat{Error}) \cong \sum_{k=0}^{73} [Error(k) - \hat{Error}(k)]^2 . \quad (4.13)$$

For quantisation of LSFs, a distortion measure with a weighting function is employed. The weighting function used in this thesis is similar to that suggested by Ramachandran, et al. [67].

$$d(LSF, \hat{LSF}) = \sum_{i=1}^{P} w(i)[LSF(i) - \hat{LSF}(i)]^2 \quad (4.14)$$

where the weighting function is defined as:

$$w(i) = \frac{1}{LSF(i) - LSF(i-1)} + \frac{1}{LSF(i+1) - LSF(i)} . \quad (4.15)$$

The purpose of this weighting is to emphasise the formant frequencies and, therefore, provide better quantising performance than the unweighted squared Euclidean distortion [67].

4.3.3 Training Algorithm

The training algorithm for the SEW codebook, the *Error* codebook and the LSF codebooks presented here is an adaptation of that described in [51]. The sets of SEW, *Error* and LSF training data are derived from approximately one hour of speech taken from a TIMIT speech database, including a large number of speakers with different accents. Each training set contains 144,000 training vectors. According to Makhoul, et al. [52] these training sets are sufficiently wide to produce good codebooks.

Initial codebooks are designed using the random entry technique described in Chapter 2, Section 2.4.4.1. The steps of the training algorithm are:

(0) *Initialisation*: Let M be the length of the training sequence of data $\{\mathbf{x}_j; j = 0,...,M-1\}$, and L, the number of vectors in the required vector quantiser. (For the 8 bit SEW codebook, L = 256; the 5 bit *Error* codebook, L = 32; the 10 bit LSF codebook, L = 1024).

Set the distortion threshold $\varepsilon \geq 0$. As the value $\varepsilon = 0$, the algorithm halts for a finite number of iterations ($m = \infty$). Here for all the codebooks, ε is assigned to be 0.001. The chosen value ε = 0.001 is an empirical method such that the number of iterations is around 15. Experiments in this work have shown that most codebooks (for SEW, *Error* and LSFs) converged in fewer than 15 iterations.

Set the initial expected distortion $D_{-1} = \infty$. (Here the value $D_{-1} = 9*10^{99}$ is considered as ∞.)

Given an initial codebook $\mathbf{y}(0) = \{\mathbf{y}_i; i = 0,..., L-1\}$.

The partition S_i is defined as $\mathbf{x} \in S_i$ if $d(\mathbf{x}, \mathbf{y}_i) \leq d(\mathbf{x}, \mathbf{y}_j)$ for all j.

Set the training step $m = 0$ and start training.

(1) *Classification*: Given $\mathbf{y}(m) = \{\mathbf{y}_i; i = 0,..., L-1\}$, find the minimum distortion partition $S_i(m) = \{\mathbf{x}_{ik}; k = 0,... M_i(m)-1\}$; $i = 0,..., L-1$ of the training sequence as: for each $j = 0,..., M-1$, compute $d(\mathbf{x}_j, \mathbf{y}_i)$, $i = 0,..., L-1$. If $d(\mathbf{x}_j, \mathbf{y}_i) \leq d(\mathbf{x}_j, \mathbf{y}_l)$ for all l, $\mathbf{x}_j \in S_i(m)$ and becomes $\mathbf{x}_{ik}$. Then compute the overall expected distortion D_m:

$$D_m = \frac{1}{M} \sum_{j=0}^{M-1} \min_{\mathbf{y} \in \mathbf{y}(m)} d(\mathbf{x}_j, \mathbf{y}). \qquad (4.16)$$

(2) *Checking for termination*: if $(D_{m-1} - D_m) / D_m \leq \varepsilon$, halt and the final quantiser is described by $\mathbf{y}(m)$. Otherwise continue to the next step.

(3) *Initial codebook updating*:

$$\mathbf{y}(m+1) = \{\mathbf{x}(S_i(m)),\ \ i = 0,\ldots,L-1\}. \tag{4.17}$$

Here $\hat{x}(S_i)$ is the Euclidean centre of gravity and defined as:

$$\mathbf{x}(S_i(m)) = \frac{1}{M_i(m)} \sum_{k=0}^{M_i(m)-1} \mathbf{x}_{ik} \tag{4.18}$$

where $M_i(m)$ is defined as the number of training vectors in the cell $S_i(m)$. If $M_i(m) = 0$, set $\mathbf{x}(S_i(m)) = \mathbf{y}_i$, the old codeword. Replace m by $m+1$ and go to Step (1).

4.3.4 Codebook Performance

As in any vector quantisation, the codebook performance plays an important role. This section presents the performance assessment of the SEW, *Error* and LSF codebooks designed in this work; and the test results are also discussed.

4.3.4.1 LSF Codebook Performance

The performance of LSF codebooks can be evaluated using the spectral distortion (SD) measure [64]. This distortion measure is defined [80] as:

Let SD_n be the spectral distortion for the frame n^{th}, where SD_n is defined as:

$$SD_n^2 = \frac{1}{F_s}\int_0^{F_s} \left[10\log_{10}(P_n(f)/\hat{P}_n(f))\right]^2 df \tag{4.19}$$

where F_s (in Hertz) is the sampling frequency of the speech, and $P_n(f)$ and $\hat{P}_n(f)$ are the LPC power spectra of the n^{th} frame. $P_n(f)$ and $\hat{P}_n(f)$ are presented as:

$$P_n(f) = 1/\left|A_n \exp(j2\pi f/F_s)\right|^2 \quad (4.20)$$

$$\hat{P}_n(f) = 1/\left|\hat{A}_n \exp(j2\pi f/F_s)\right|^2 \quad (4.21)$$

where $A_n(z)$ and $\hat{A}_n(z)$ are the unquantised and quantised LPC polynomials respectively of the frame.

It is believed that quantisation of the LSFs is transparent if the following three requirements are satisfied [80]:

1. the average distortion is about 1 dB;
2. the number of outlier frames having spectral distortion above 2 dB is less than 2%; and
3. there are no outlier frames with spectral distortion greater than 4 dB.

For assessment of the LSF VQ, 200 seconds of speech (which was not included in the training set) was used for testing. The speech was analysed using a Hamming window of 20ms; thus there were 10000 LSF vectors for testing. For comparison, the SD was calculated for this 30 bits/frame LSF split VQ and also for the 34 bits/frame LSF scalar quantiser used in the US 1016 Federal Standard CELP-4.8kb/s [28].

The results are given in Table 4.3. From Table 4.3, it can be seen that the average SD and the percentage of outliers with SD greater than 2 dB for the 30 bit LSF split-VQ is less than that resulting from the 34 bit LSF SQ (used in the US 1016 Federal Standard CELP-4.8kb/s). However, the percentage of outliers with SD greater than 4 dB of the 30 bit LSF split VQ is higher than that of the 34 bit LSF SQ. Thus, it can be concluded that the performance of the 30 bit LSF split-VQ is equivalent to or better than the performance of the 34 bit LSF SQ.

	Avg. SD (dB)	Outlier (%) 2-4 dB	Outlier (%) > 4 dB
30 bit LSF Split-VQ	1.2	4.8	0.04
34 bit LSF SQ (US FS1016)	1.4	11.0	0.01

Table 4.3 Spectral Distortion Performance of the 30 bit LSF Split-VQ and the 34 bit LSF SQ (US 1016 Federal Standard)

4.3.4.2 SEW and *Error* Codebooks Performance

Since the aim of both the SEW and *Error* codebooks is to provide codebook vectors which best match the unquantised waveforms, the performance of these codebooks can be evaluated using the Average SNR measure (in dB). The average SNR expression for the evaluation of the SEW codebook across N frames is given below (the expression for the *Error* codebook is similar).

$$Av.SNR = 10\log_{10}\left\{\frac{1}{N}\sum_{n=1}^{N}\frac{\sum_{k=0}^{p_n-1}\left|SEW_n(k)\right|^2}{\sum_{k=0}^{p_n-1}\left|SEW_n(k)-\hat{SEW}_n(k)\right|^2}\right\} \qquad (4.22)$$

where $SEW_n(k)$ and $\hat{SEW}_n(k)$ are the unquantised and quantised SEW for the n^{th} frame; and p_n is the pitch length (in samples) for the n^{th} frame.

For assessment, a set of 14 TIMIT speech files was used. The Average SNR for the SEW codebook is 9.5 dB, and for the *Error* codebook, 7.7 dB. Figure 4.3 and Figure 4.4 show the match between the unquantised SEWs and quantised SEWs; and between the unquantised *Error*s and the quantised *Error*s, respectively. These results indicate a reasonable match between the quantised and unquantised SEW (*Error*) vectors, however, the final assessment must be in terms of the overall quality of the coded speech. This is presented in the following chapters.

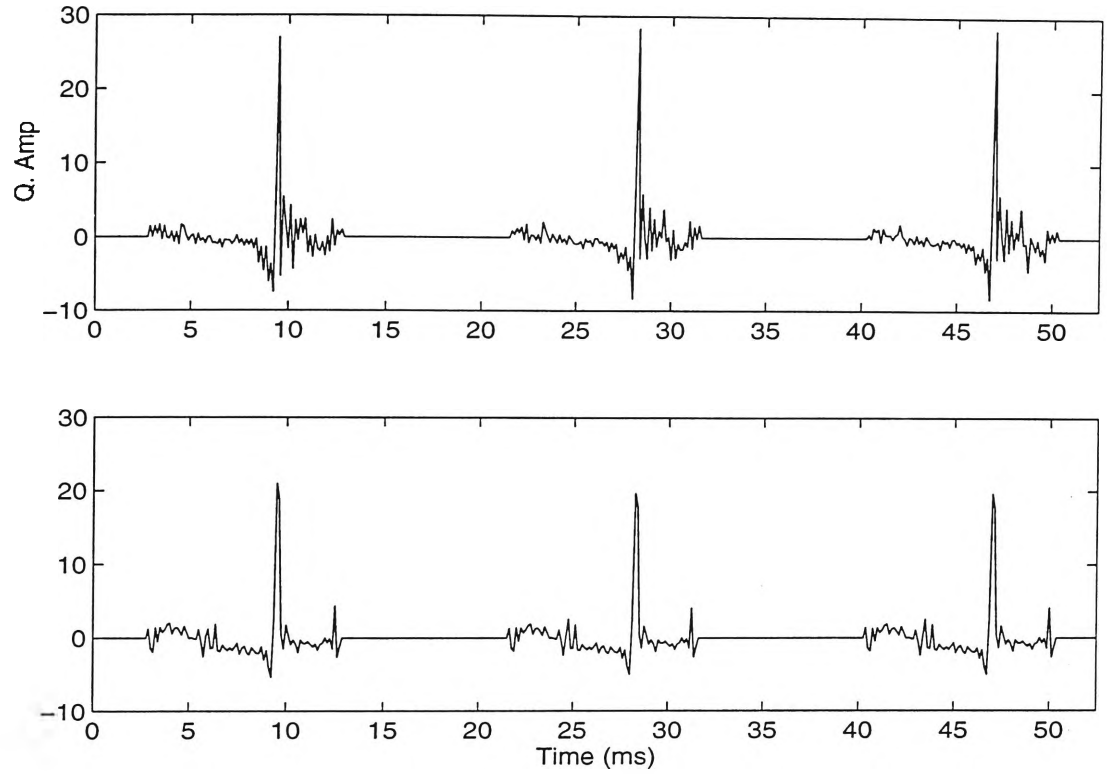


Figure 4.3. An Example of the Matching Between Unquantised SEW Vectors (top) and Quantised SEW Vectors (bottom)

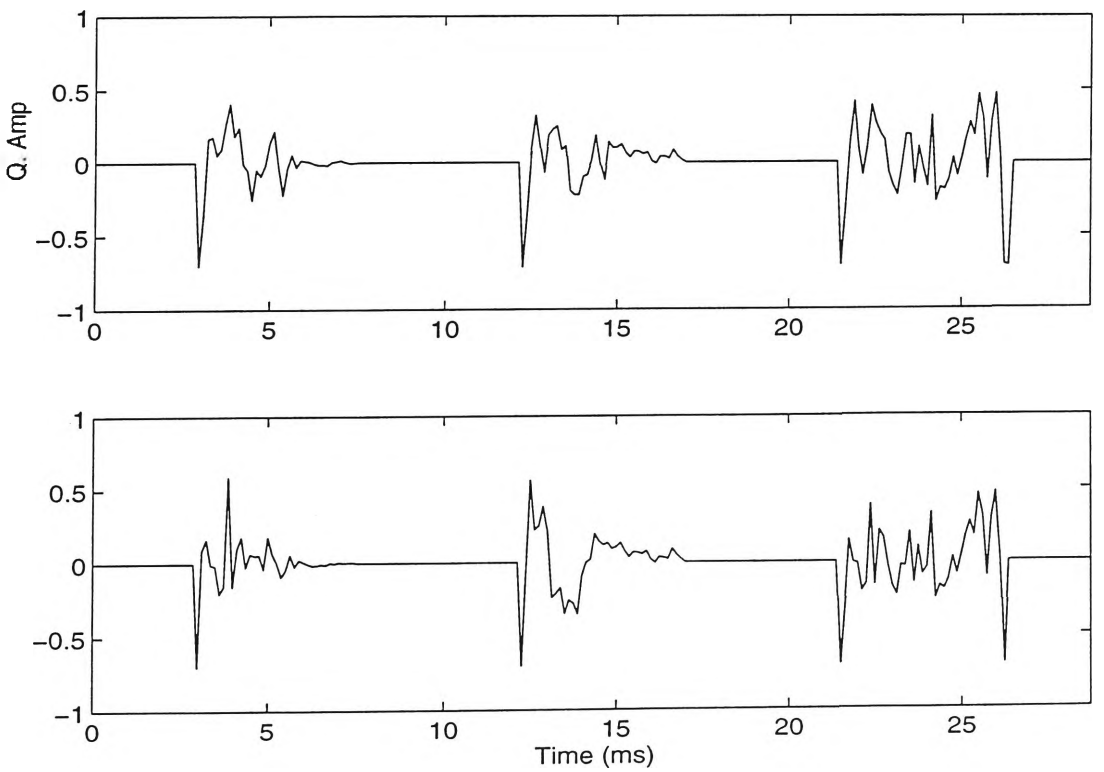


Figure 4.4 An Example of the Matching Between Unquantised *Error* Vectors (top) and Quantised *Error* Vectors (bottom)

4.4 Experimental Results

This section presents the experimental results of the Unity Magnitude Coder and the Errored Magnitude Coder.

For a coder to be subjectively tested fully, both quality and intelligibility tests are required to be conducted under numerous conditions, such as quiet with a microphone, modern office, airplane, etc., which are chosen based on availability and relevance to the civilian and military services [23]. These tests were only performed for coders that were candidates for the US Government Standard, and were conducted by either the US Government or US Department of Defense [23,7,40]. In this work, due to the time limitations and certain conditions for the Masters degree, the MPW coders were tested using the Mean Opinion Score in the quiet condition with a microphone. First of all, it should be noted that the 30 bit split LSF VQ has a performance equivalent to that of the 34 bit LSF SQ of the US 1016 Federal Standard CELP-4.8kb/s. This ensures that the performance comparison between the MPW coders and the US 1016 Federal Standard CELP-4.8kb/s is reliable.

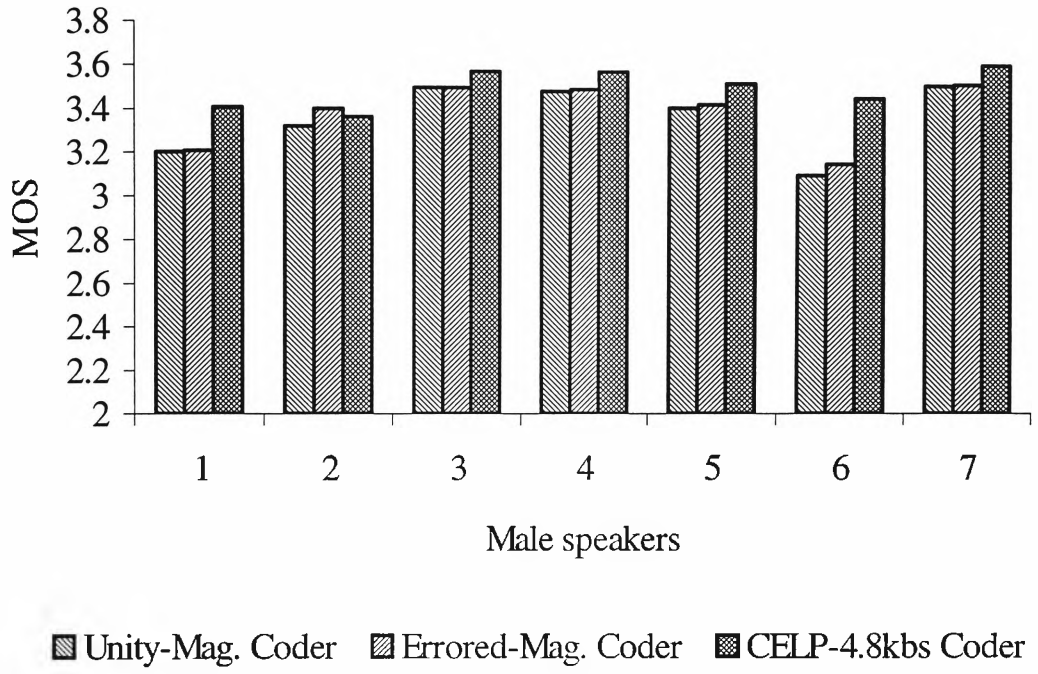
The MPW coders were tested using the MOS described in Chapter 2. Fourteen TIMIT standard speech files consisting of seven male and seven female speakers were used as references, and the tests were carried out using a large number of well-trained listeners. During the test, the MPW coders were compared with the CELP-4.8kb/s Coder (similar to the US 1016 Federal Standard). The results are presented in Table 4.4. These results are also shown in the bar charts of Figure 4.5.

From Table 4.4 and Figure 4.5 it can be seen that the MOS for the speech files coded by both MPW coders is close to the MOS for those coded by the CELP-4.8kb/s coder. The average MOS for the Unity Magnitude Coder, the Errored Magnitude Coder and the CELP-4.8kb/s Coder are 3.415, 3.433 and 3.536

respectively. The MOS for both MPW coders is within 0.1 lower than the MOS for the CELP-4.8kb/s coder. The MOS for the Unity Magnitude Coder is 0.02 lower than that for the Errored Magnitude Coder. Note that the tests conducted by the US Government in 1994 have shown that the MOS for the US 1016 Federal Standard CELP-4.8kb/s in a quiet condition is 3.59 [23,40,43]. These results indicated that the performance of both MPW coders was equivalent to that of the US 1016 Federal Standard CELP-4.8kb/s. In addition, the quantisation of the *Error* provided a small improvement in the quality of the coded speech.

Testing speech files	Unity-Mag. Coder	Errored-Mag. Coder	CELP-4.8kb/s Coder
male 1	3.200	3.210	3.410
male 2	3.320	3.400	3.360
male 3	3.500	3.500	3.570
male 4	3.48	3.490	3.570
male 5	3.410	3.420	3.520
male 6	3.100	3.150	3.450
male 7	3.510	3.520	3.610
female 1	3.410	3.420	3.530
female 2	3.450	3.450	3.420
female 3	3.500	3.520	3.650
female 4	3.500	3.500	3.590
female 5	3.510	3.530	3.630
female 6	3.470	3.500	3.670
female 7	3.4500	3.460	3.530
average results	3.415	3.433	3.536

Table 4.4 The MOS Test Results of the Unity Mag. Coder and the Errored Mag. Coder Compared with the CELP-4.8kb/s Coder (US 1016 Federal Standard)



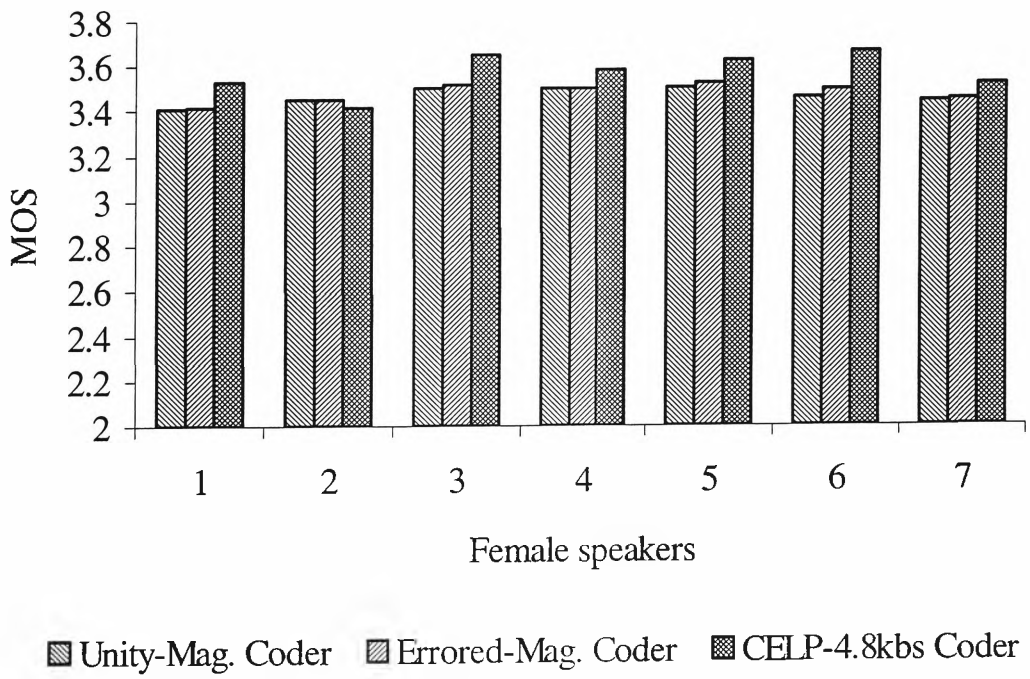


Figure 4.5 Bar Charts of the MOS Test Results for the Unity Mag. Coder and the Errored Mag. Coder Compared with the CELP-4.8kb/s Coder (US 1016 Federal Standard)

4.5 Conclusion

This chapter has described two Open Loop Quantisation schemes for prototype waveforms. In these quantisation schemes, the PWs were extracted in both voiced and unvoiced speech, and decomposed into a SEW and REW. For each frame of 25ms, the SEW was formed as a 'mean' PW of the ten extracted PWs. The REW was regarded as the remainder of the PW. The quantisation requirements of the SEW and REW were examined in the Unity Magnitude Quantisation scheme and Errored Magnitude Quantisation scheme.

The Unity Magnitude Quantisation quantised the SEW using an 8 bit VQ on both magnitude and phase spectra; while the REW was not quantised. At the receiver, the REW phase spectra was derived from a Gaussian noise source, and the REW magnitude spectra was recovered by using SEW behaviours and using the unity approximation of the magnitude of normalised PWs.

As the degradation of the coded speech quality was insignificant when transmitting the gain term at the update rate of 80Hz rather than 120Hz, the Errored Magnitude Quantisation scheme was introduced. With the aim of making improvements to the Unity Magnitude Quantisation scheme, this scheme quantised the *Error*, which is the difference between the actual magnitude of the normalised PW and unity, using a 5 bit VQ with an update rate of 40Hz. At the receiver, the REW magnitude was derived from *Error* and unity.

For these quantisation schemes to be successful, codebooks for the SEW, REW/*Error* were designed using a simple solution. The SEW codebook is of 8 bits and contains both SEW magnitude and phase spectra. The 5 bit *Error* codebook consists of the *Error* magnitude only. The tests have indicated a good match between the unquantised vector and the quantised vectors, however, the performance of the codebooks must be proved by the quality of

the coded speech. The LSF codebooks were trained using a long sequence of data. The test results show that the performance of these codebooks is better than or equivalent to the 34 bits/frame LSF SQ (US 1016 Federal Standard CELP-4.8kb/s coder).

The MOS tests show that the performance of the Unity Magnitude Coder and Errored Magnitude Coder was close to that of the US 1016 Federal Standard CELP-4.8kb/s. The Errored Magnitude Quantisation aimed to improve the speech quality; its MOS was slightly higher than that for Unity Magnitude Quantisation.

The coded speech quality was mainly dependent on the SEW. At the rate of 2.4kb/s it was effective to quantise the SEW using an 8 bit VQ on both the magnitude and phase spectra with an update rate of 40Hz. The REW, which determines the naturalness and the dynamics of the coded speech, can be effectively quantised at low bit rates by transmitting the error of the unity magnitude approximation.

Chapter 5: Multi-Prototype Waveform Analysis-by-Synthesis Coding

In the previous chapter, the MPW Open Loop Quantisation coding technique has been shown to be capable of producing good quality speech at bit rates as low as 2.4kb/s. The coded speech quality was close to that produced by the US 1016 Federal Standard CELP-4.8kb/s. This coding technique and other techniques such as those proposed in [40-44] are based on the SEW/REW decomposition paradigm, whereby the prototype waveform quantisation is performed by decomposing the PW into a SEW and REW. The SEW and REW were then quantised separately. There are certain differences between them; primarily in the definition and quantisation of the SEW/REW. However, both of them code the PWs based on a direct coding architecture which did not make full use of the available bit rate, thus it is necessary to code the PWs more efficiently. This can be achieved by using an analysis-by-synthesis architecture which offers improved prototype waveform quantisation performance. This chapter proposes a coding technique in an analysis-by-synthesis architecture for prototype waveforms. Four Analysis-by-Synthesis based MPW coders operating at bit rates of 2.4kb/s, whereby the PW is presented either in the residual or in the speech domain, called the Unity

Magnitude Residual Coder, Unity Magnitude Speech Coder, Errored Magnitude Residual Coder and Errored Magnitude Speech Coder, are examined and experimental results are presented.

5.1 Motivation for MPW Analysis-by-Synthesis Coding

Before detailed discussion on the motivation for using the MPW Analysis-by-Synthesis coding technique, it is worth considering a MPW coding system without quantisation (as shown in Figure 5.1).

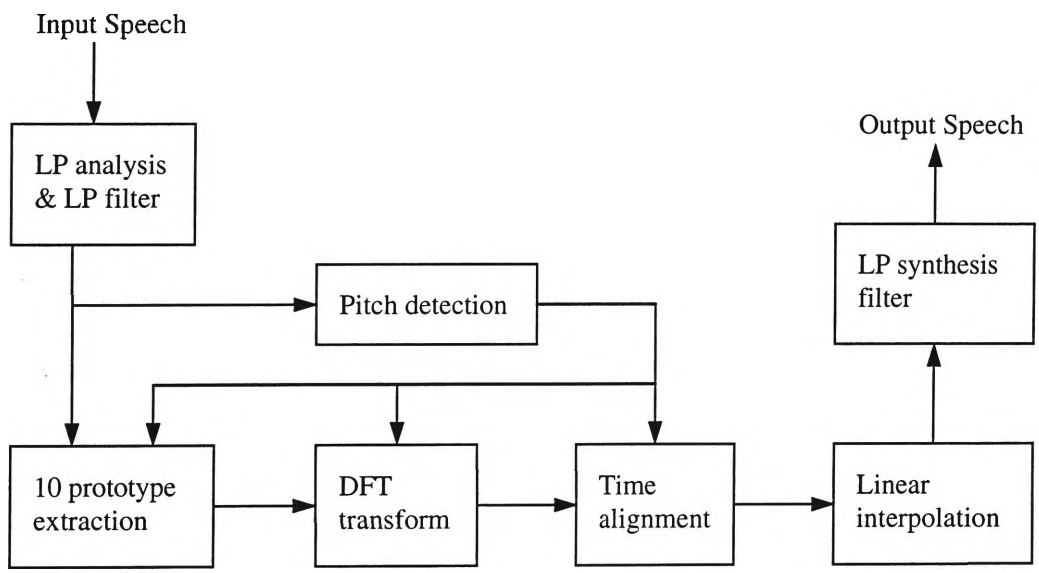


Figure 5.1 Prototype Waveform Coding System Without Quantisation

This coder extracts 10 PWs each frame (i.e., at an update rate of 400Hz). The extracted PWs are DFT transformed and then time aligned to ensure that the current PW aligns with the previous PW. To obtain the whole residual, these PWs are continuously interpolated in the DFT domain. The residual signal is then filtered using a LP synthesis filter to produce coded speech. As can be seen (from Figure 5.2), in this case, the reconstructed speech is nearly

identical to the original signal. An important feature of the prototype waveform interpolation coding technique is that the speech is represented as a concatenation of individual PWs. From this discussion, it is possible to draw the conclusion that if the extracted PWs were quantised and could be recovered perfectly then, as a result, the reconstructed speech would be perceptually identical to the input speech except for nonsynchronisation between them [34,40].

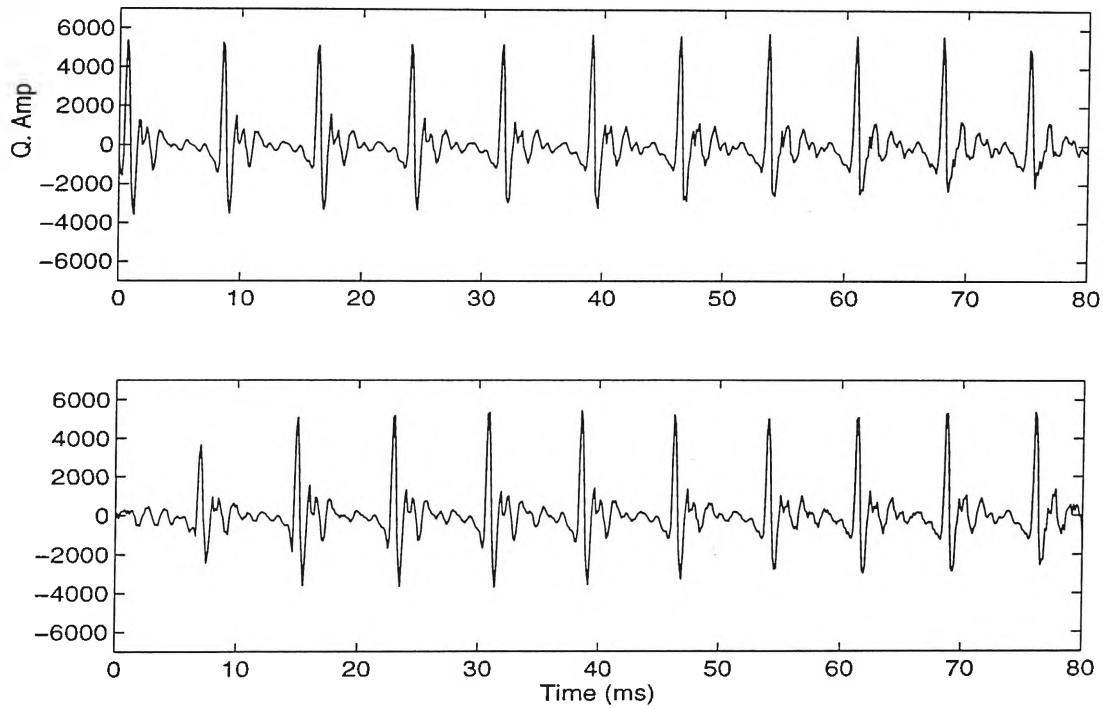


Figure 5.2 Input Speech (top) and Output Speech (bottom) of the MPW Coder in Figure 5.1

CELP techniques can provide good quality speech at the rate of 4.8kb/s. The advantage of CELP is that the excitation signal is chosen from the codebook based on the synthesised speech it produces. The CELP coding algorithm can be regarded as choosing an excitation codebook vector which when filtered by the cascaded LTP/LPC filters, best matches the input speech segment. The theory of this chapter is that in a similar manner to CELP, MPW coders could provide better quality synthesised speech, if designed using analysis-by-

synthesis architecture. The basic feature of prototype waveform coding is that it processes speech via individual PWs. This feature is also an advantage. It has already been shown that unquantised prototype waveforms can produce high quality speech. Analysis-by-synthesis on the prototype waveform should also produce high quality speech. Therefore, an MPW A-by-S coder should include this feature. The basic coding algorithm of the proposed MPW analysis-by-synthesis coding technique in this work can be regarded as choosing a SEW codebook vector which, when added with the reconstructed REWs, best matches the extracted PWs each frame. Details of the technique are discussed in the following sections.

5.2 A-by-S Unity Magnitude Residual Quantisation

This prototype waveform quantisation scheme has an analysis-by-synthesis architecture and is based entirely on the assumption that the normalised PW magnitude is unity. The REW magnitude is the remainder of the SEW magnitude and unity. Being a random noise, the REW phase is not quantised; and thus it is derived from a Gaussian noise at the decoder. The difference between the MPW Open Loop Quantisation technique discussed in Chapter 4 and this technique is that the PW is not decomposed but is considered as a combination of the SEW and REW.

5.2.1 Unity Magnitude Residual Coder

Basically, the architectures of the MPW Analysis-by-Synthesis coders are similar to MPW Open Loop coders. The difference between them is the SEW codebook searching algorithms. This section presents a brief description of a MPW A-by-S coder: the Unity Magnitude Residual Coder.

Encoder

The architecture of the Unity Magnitude Residual Encoder is given in Figure 5.3. For each 25 ms frame, the 10^{th} order LPC coefficient estimation and LSF quantisation are performed in the same manner as that described in Section 3.1.2. The pitch is estimated using the technique described in Section 3.2.

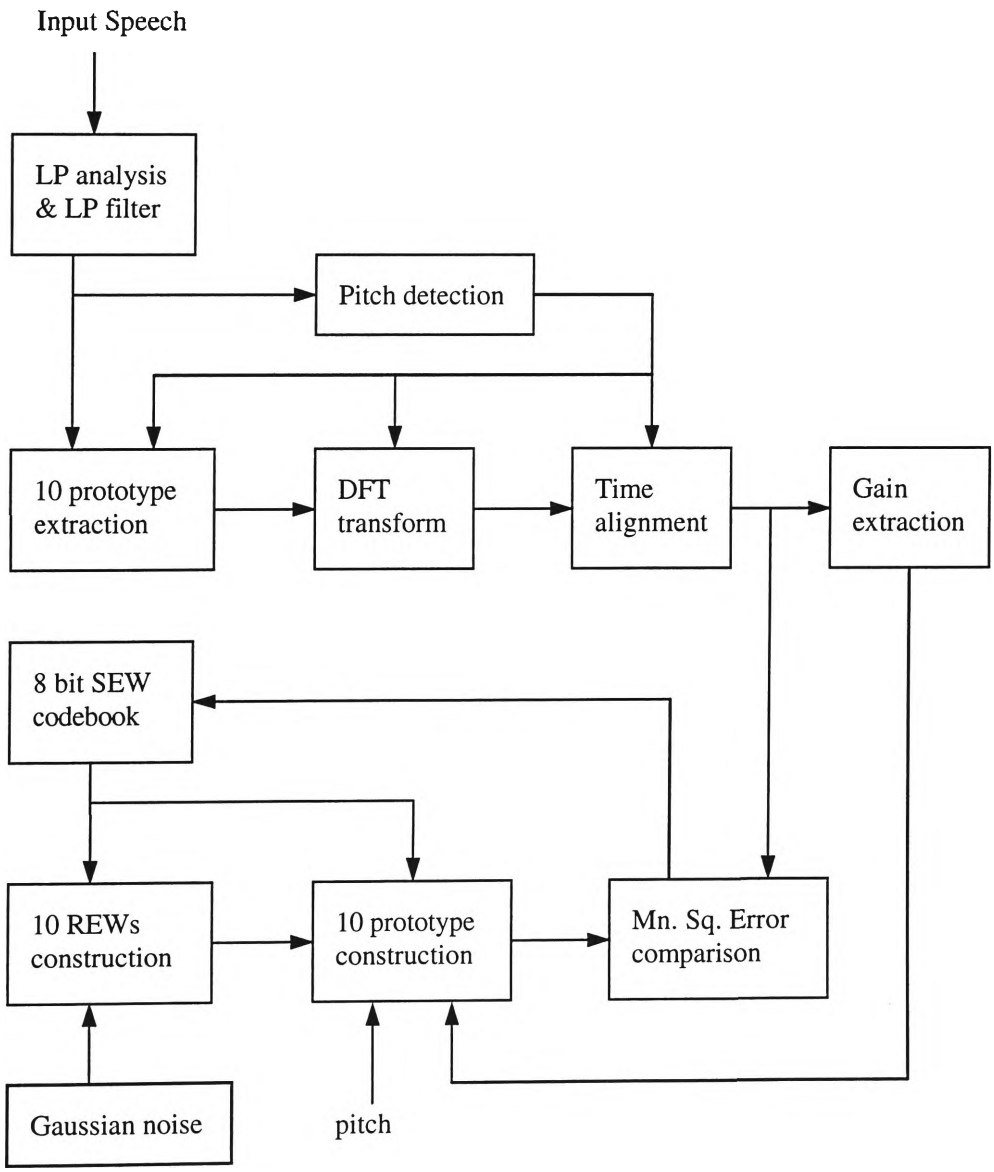


Figure 5.3 Unity Magnitude Residual Encoder Architecture

The PW is extracted from the residual signal at the rate of 400Hz using the technique discussed in Section 3.3. For quantisation, the PW is DFT transformed and time aligned. The extracted PWs are quantised using a gain/shape VQ. The gain terms are extracted and quantised using a logarithmic 5 bit SQ with an update rate of 120Hz. The PWs are normalised and then quantised. For quantisation, rather than being decomposed into a SEW and a REW, the PW is regarded as a combination of them. The unity magnitude approximation allows the REW to be recovered according to the SEW behaviours at the decoder. The 8 bit SEW codebook is searched on the basis of best matching between the extracted PWs and the candidate PWs constructed from the SEW codebook vector. (Details of the SEW codebook search will be discussed in the following section.)

Decoder

The block diagram of the Unity Magnitude Residual Decoder is shown in Figure 5.4. It can be viewed as a part of the Unity Magnitude Residual Encoder. Once the SEW is decoded, it is upsampled from 40Hz to 400Hz. Ten REW magnitudes are reconstructed from these ten SEW magnitudes and unity using Equation (4.5). The REW spectral phase is derived from a Gaussian noise source. The sum of the SEW and REW constructs the normalised PW. The ten complete PWs are obtained by multiplying the normalised PWs with the associated decoded gain terms.

The ten PWs are then time aligned. To obtain the complete residual signal, these PWs are continuously interpolated in the DFT domain in the same manner as that described in Section 3.5. The residual signal is then filtered by using the LP synthesis filter to produce the speech signal. For perceptual speech quality enhancement, a postfilter [75-77] is used in cascade with the LP synthesis filter.

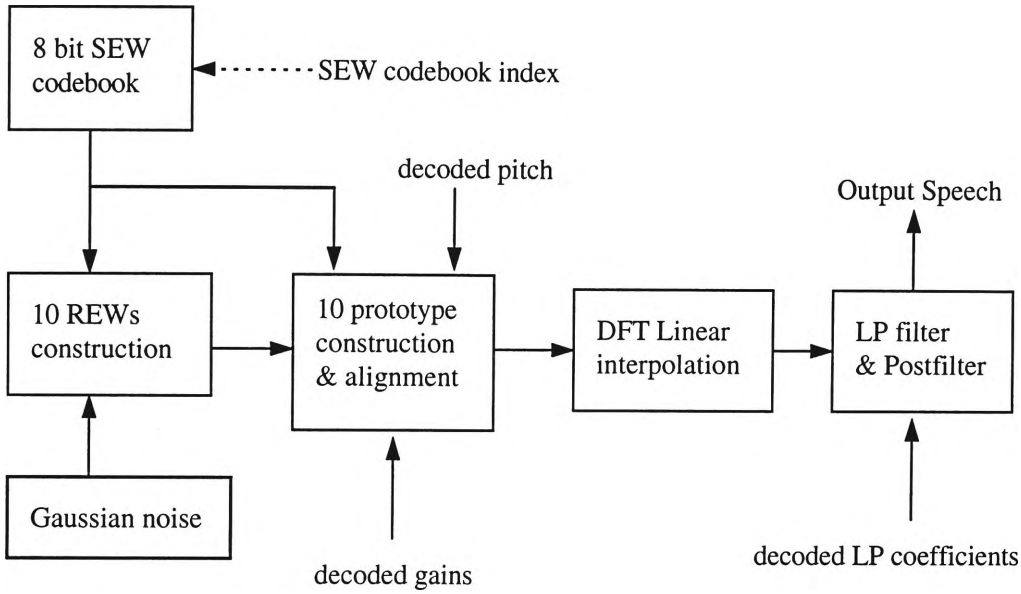


Figure 5.4 A-S Unity Magnitude Residual MPW Decoder Architecture

The bit allocation for this coder is shown in Table 5.1. The total number of bits is 60 bits per 25ms frame. With this bit allocation plan, the total bit rate is 2.4kb/s.

Parameter	Number of bits	Update rate
LSF	30	40 Hz
pitch	7	40 Hz
gain	5	120 Hz
SEW	8	40 Hz

Table 5.1 Bit Allocation for the Unity Magnitude Residual Coder

5.2.2 Codebook Search

In CELP, the excitation vector is chosen from the fixed codebook based on the best possible match between the synthesised and input speech. Similarly for this coder, in the residual domain, the SEW codebook searching algorithm is regarded as choosing a SEW which, when upsampled to 400Hz and added to the REW, can best match the incoming residual PWs extracted at a rate of 400Hz.

As the PW can be decomposed into a SEW and a REW, the sum of the SEW and REW must be the PW. In this technique, the REW magnitude is derived from the SEW magnitude based on the unity magnitude assumption, and the REW phase is taken from a Gaussian noise source. Although information about the REW phase is not transmitted, the use of the REW phase is not only for making the complete candidate PW but also for the codebook search to be more effective. The reason for this is explained in the following section.

5.2.2.1 Role of the REW Phase Spectra

The role of the REW phase spectra in the SEW codebook search is illustrated in Figure 5.5, which shows: an input signal (A), a codebook vector (B), a random noise (C), and the codebook vector added with the random noise (D). The waveform of the signals is randomly assumed as shown in the figure. The input signal is a sinusoidal waveform, and the codebook vector is effectively a quantised sinusoidal signal. It can be seen that the waveform of the codebook vector after random noise was added looks smoother and closer to the waveform of the input signal. Clearly, it can be seen that the use of random noise smooths the codebook vector such that it better matches the input signal.

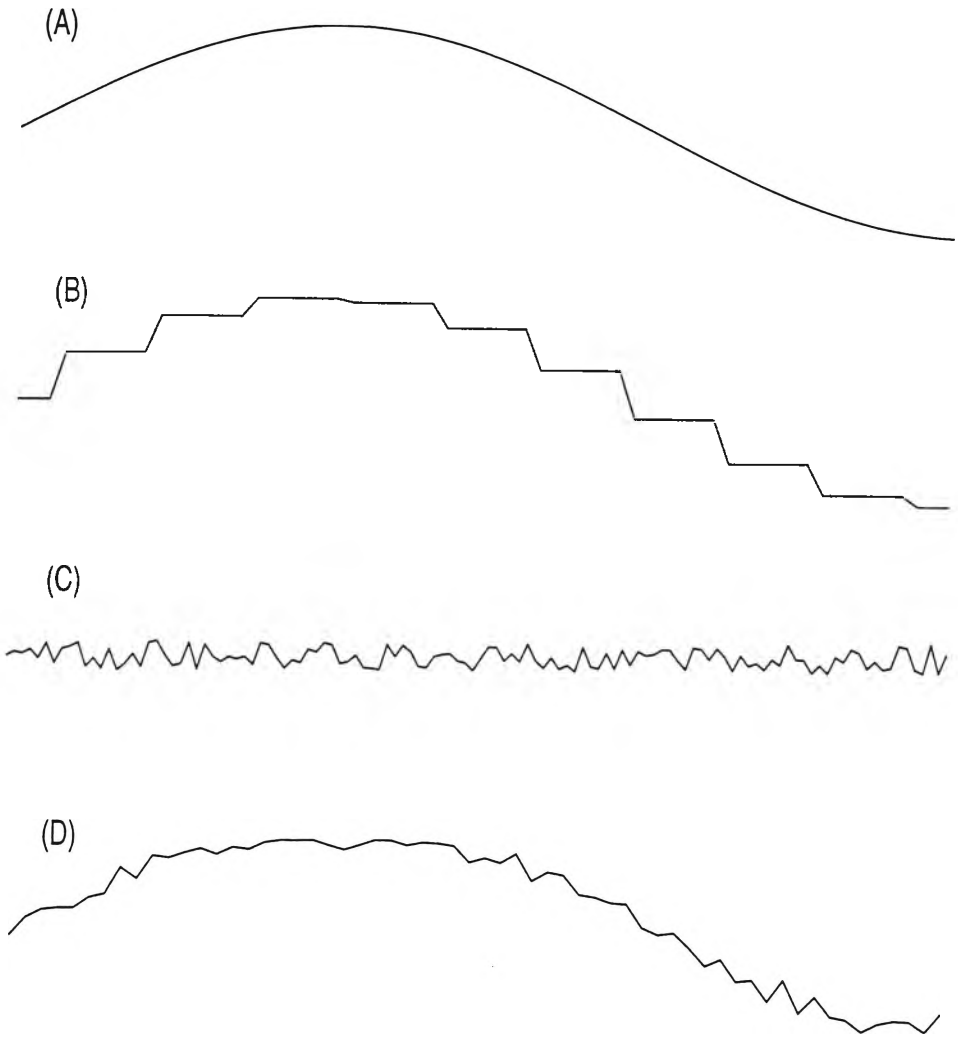


Figure 5.5 Input Signal (A), Codebook Vector (B), Random Noise (C), Codebook Vector With Random Noise (D)

To prove the above, let us consider Figure 5.6, which shows an example of a residual PW extracted from a segment of voiced speech spoken by a male speaker and its associated signals: the candidate PW, the chosen SEW codebook vector, and the REW. The REW magnitude is constructed according to the SEW magnitude, and the REW phase is Gaussian noise. As expected, it is clear that the candidate PW (the SEW codebook vector after the REW added) is better matched to the extracted PW than the SEW alone. As shown in Figure 5.7 this discussion is also valid for the case of unvoiced speech segments.

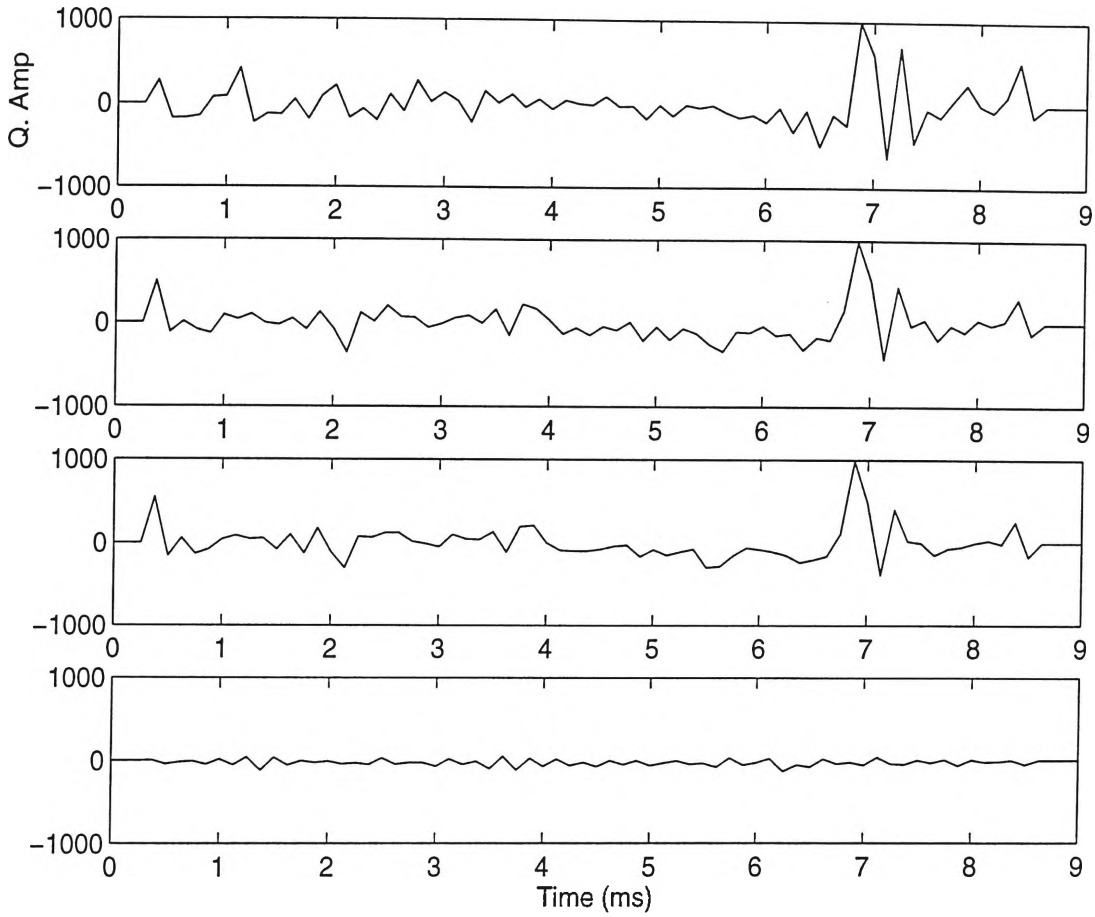


Figure 5.6 From the Top to the Bottom: Extracted PW, Candidate PW, SEW Codebook Vector, and REW for a Voiced Speech Segment

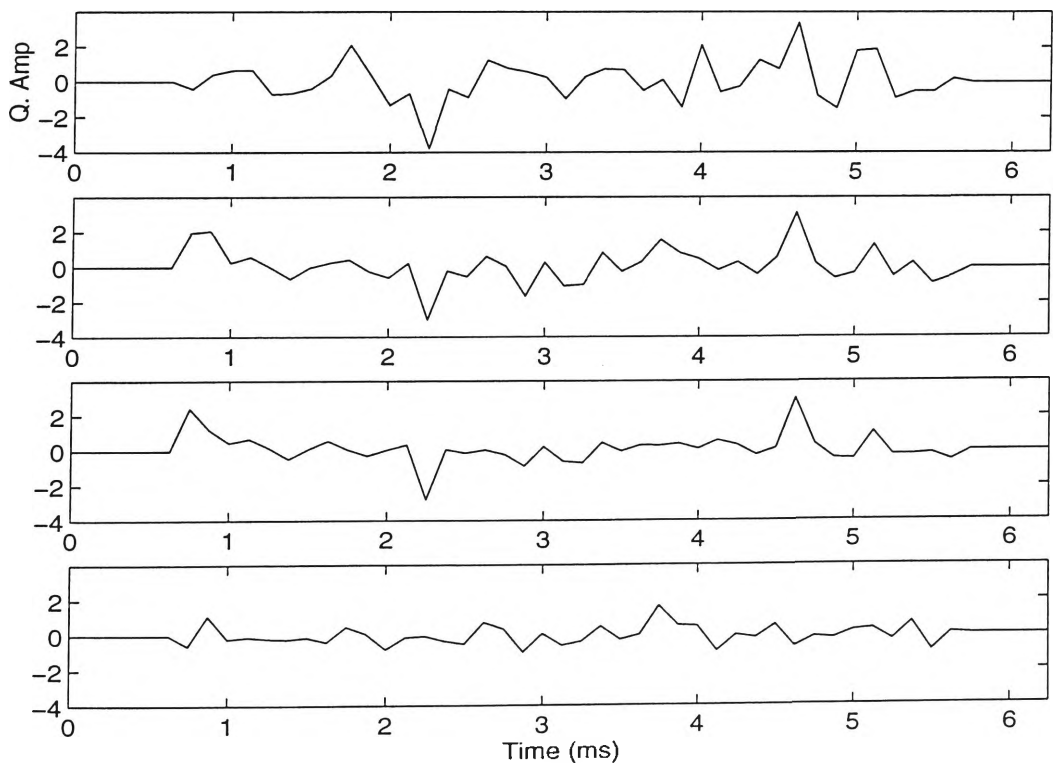


Figure 5.7 From the Top to the Bottom: Extracted PW, Candidate PW, SEW Codebook Vector, and REW for a Unvoiced Speech Segment

Furthermore, the use of Gaussian noise as a REW phase in the SEW codebook search is necessary, as without Gaussian noise the candidate PW cannot be completed. To evaluate the role played by Gaussian noise, it is worth considering a special case wherein the REW spectral phase is kept identical to the SEW phase spectra:

$$REWpha_m(k) = SEWpha_m(k) \qquad for\ k = 0,\ldots, p_m - 1. \tag{5.1}$$

The REW magnitude spectra is constructed as:

$$REWmag_m(k) = 1.0 - SEWmag_m(k) \quad for\ k = 0,\ldots, p_m - 1. \tag{5.2}$$

Thus, the constructed PW would be formed as:

$$X_m(k) = REW_m(k) + SEW_m(k) \quad for\ k = 0,\ldots, p_m - 1. \tag{5.3}$$

The real value and the imaginary value of the candidate PW would be:

$$\begin{aligned}\mathrm{Re}[X_m(k)] = {} & REWmag_m(k)\cos(REWpha_m(k)) \\ & + SEWmag_m(k)\cos(SEWpha_m(k)\end{aligned} \tag{5.4}$$

$$\begin{aligned}\mathrm{Im}[X_m(k)] = {} & REWmag_m(k)\sin(REWpha_m(k)) \\ & + SEWmag_m(k)\sin(SEWpha_m(k)\end{aligned} \tag{5.5}$$

$$for\ k = 0,\ldots p_m - 1;\ \ m = 1,\ldots,10.$$

According to Equations (5.2) and (5.3), Equations (5.4) and (5.5) can be rewritten as:

$$\mathrm{Re}[X_m(k)] = [1 - SEWmag_m(k) + SEWmag_m(k)]\cos(SEWpha_m(k)) \tag{5.6a}$$

$$\mathrm{Im}[X_m(k)] = [1 - SEWmag_m(k) + SEWmag_m(k)]\sin(SEWpha_m(k)). \tag{5.6b}$$

The candidate PWs would be formed as follows:

$$X_m(k) = \cos[SEWpha_m(k)] + j\sin[SEWpha_m(k)]. \tag{5.7}$$

Clearly from Equation (5.7), the candidate PW is now only dependent on the SEW phase spectra which is chosen from the SEW codebook. In this case the speech is purely voiced and has no unvoiced component. The choice of an identical REW phase spectra to the SEW phase spectra is a violation of the fact that the REW phase spectra is random. Hence, this leads to incorrect construction of the candidate PWs. From this, it is possible to conclude that although the REW phase spectra is a Gaussian noise, it cannot be ignored in the construction of candidate PWs for the SEW codebook search.

5.2.2.2 Codebook Search

The codebook search algorithm can be described as minimising the total mean squared error between the extracted PWs and the candidate PWs constructed from the SEW codebook vector. For codebook searching, the extracted PWs are aligned with the standard vector (see Section 4.3.1). Since there are ten PWs in each frame, the total mean squared error can be regarded as the total of the ten mean squared errors between ten extracted PWs and ten candidate PWs. The formula for this is as follows:

$$E^{(c)} = \frac{1}{10}\sum_{m=1}^{10} E_m^{(c)} \qquad c = 0,\ldots,255 \tag{5.8}$$

where c is the SEW codebook index, $E^{(c)}$ and $E^{(c)}_m$ are the total and the individual mean squared error respectively. Generally, $E^{(c)}_m$ can be calculated by the formula:

$$E_m^{(c)} = \frac{1}{P_m}\sum_{k=0}^{P_m-1} f(k)\left|PW_m(k) - G_m^{(c)} X_m(k)\right|^2 \qquad for\ m = 1,\ldots,10 \tag{5.9}$$

where $PW_m(k)$ and $X_m(k)$ are the incoming PW and the candidate PW respectively in the DFT domain; $G_m^{(c)}$ is the code gain term and $f(k)$ is a

weighting function which depends on $PW_m(k)$. In Equation (5.9), the $||^2$ term is the norm of the expression such that this equation can be rewritten as:

$$E_m^{(c)} = \frac{1}{p_m}\sum_{k=0}^{p_m-1} f(k)[PW_m(k) - G_m^{(c)} X_m(k)][PW_m(k) - G_m^{(c)} X_m(k)]^* \qquad (5.10)$$

where $[]^*$ is the conjugate of [].

The weighting function is used to improve quantisation performance. However, for simplification, in this work the weighting function *f(k)* is not used (or assumed as unity). Thus, the expression for $E^{(c)}$ can be written as:

$$E^{(c)} = \frac{1}{10}\sum_{m=1}^{10}\left\{\frac{1}{p_m}\sum_{k=0}^{p_m-1}\left|PW_m(k) - G_m^{(c)} X_m(k)\right|^2\right\}. \qquad (5.11)$$

The aim of the quantisation process is to find the minimum of $E^{(c)}$; thus the gain term $G_m^{(c)}$ generally could be found by setting the derivative $\partial E^{(c)} / \partial G_m^{(c)} = 0$ in Equation (5.11) such that:

$$G_m^{(c)} = \frac{\sum_{k=0}^{p_m-1}\mathrm{Re}[PW_m^*(k)X_m(k)]}{\sum_{k=0}^{p_m-1}\left|X_m(k)\right|^2} \qquad for\ m = 1,...10 \qquad (5.12)$$

and the minimum of $E^{(c)}$ becomes the maximisation of the expression:

$$E'^{(c)} = \frac{1}{10}\sum_{m=1}^{10}\frac{1}{p_m}\left\{\frac{\left(\sum_{k=0}^{p_m-1}\mathrm{Re}[PW_m^*(k)X_m(k)]\right)^2}{\sum_{k=0}^{p_m-1}\left|X_m(k)\right|^2}\right\}. \qquad (5.13)$$

This maximisation is searched across the codebook. However, these formulas are complicated and should not be used. In this work, it is worth utilising the feature of the MPW coders. It should be noted that the SEW codebook is

trained by a set of the SEW training data which were derived from the normalised PWs. For each SEW codebook vector, ten PWs are constructed based on the assumption that the magnitude spectra of a normalised PW is flat and equal to unity. Thus, independently from the codebook vector, the code gain term can be calculated from the extracted PW using the expression:

$$G_m = \frac{1}{p_m} \sum_{k=0}^{p_m-1} [PW_m(k) PW_m^*(k)] \qquad for\ m = 1,...,10 \tag{5.14}$$

and the expression for the individual mean squared error is:

$$E_m^{(c)} = \frac{1}{p_m} \sum_{k=0}^{p_m-1} \left| PW_m(k) - G_m X_m(k) \right|^2 . \tag{5.15}$$

The quantisation process is to search across the SEW codebook such that $E^{(c)}$ calculated from the following equation is minimised:

$$E^{(c)} = \frac{1}{10} \sum_{m=1}^{10} \left[\frac{1}{p_m} \sum_{k=0}^{p_m-1} \left| PW_m(k) - G_m X_m(k) \right|^2 \right]. \tag{5.16}$$

As the SEW codebook search is not dependent on prototype waveform decomposition, the codebook vector can be either defined as the mean PW of the extracted PWs in each frame or defined as that obtained by means of low-pass filtering the extracted PWs.

5.3 A-by-S Unity Magnitude Speech Quantisation

This section describes a MPW analysis-by-synthesis quantisation scheme, which is similar to the scheme described in Section 5.2. However, the difference between them is the SEW codebook searching algorithm. Thus the encoder is different from the Unity Magnitude Residual Encoder, while the decoder is the same as the Unity Magnitude Residual Decoder. The SEW

codebook search is based on the best matching of the extracted PW and the candidate PW in the speech domain rather than in the residual domain (used in the A-by-S Unity Magnitude Residual Quantisation Scheme). Such a prototype waveform quantisation scheme is more complex. In the speech domain, however, a weighting filter can be employed to exploit the masking behaviour of human hearing during the SEW codebook search. The role of the weighting filter is to redistribute the speech power away from the formants, such that the high energy bands are de-emphasised and the low energy bands are emphasised [81]. Thus, the quantisation performance could be more effective.

This section examines the use of a weighting filter in the MPW analysis-by-synthesis coding. The architecture of the Unity Magnitude Speech Encoder is similar to the architecture of the Unity Magnitude Residual Encoder, however, it has two DFT weighted synthesis filters for synthesising the residual PWs. The bit allocation for this coder is the same as that of the Unity Magnitude Residual Coder given in Table 5.1.

5.3.1 Weighting Synthesis Filter

A weighting synthesis filter, generally, is an IIR filter and is defined as:

$$H_{\gamma(IIR)}(z) = \frac{1}{A(z/\gamma)} = \frac{1}{1-\sum_{i=1}^{p} a(i)\gamma^i z^{-i}} \qquad 0 \le \gamma \le 1 \qquad (5.17)$$

where $A(z)$ is the standard LPC analysis filter (FIR filter) and parameter γ is for controlling the energy in the formants and normally given a value of 0.8 or 0.9. Further details about this can be found in [59]. Since this filter is IIR, even if the input sequence is identically zero, the filter would produce a non-zero output sequence (given a non-zero initial condition). For the codebook search, this 'zero response' is necessarily removed before the searching process.

Rather than a convolution in the time domain, the filtering of the PW in the DFT domain, simply, is a vector multiplication of the DFT coefficients of the PW and the DFT of the impulse response of the LP synthesis filter. One mechanism for this perceptually weighted filtering is to use the DFT of the truncated IIR LP synthesis filter such that:

$$PW_m^{(S)}(k) = H_{\gamma(IIR)}(k)PW_m(k) \qquad for\ k = 1,\ldots p_m - 1 \qquad (5.18)$$

where $PW_m^{(S)}(k)$ denotes the DFT of the prototype waveform in the speech domain. However, this truncation would lead to distortion and attenuation. Another mechanism is to implement a weighting LP synthesis filter via a FIR filter in the DFT domain. Such a mechanism can avoid the distortions caused by truncation.

The DFT of the impulse response of an IIR filter, $H_{\gamma(IIR)}(k)$ is related to the impulse response $h_{\gamma(IIR)}(n)$ as:

$$H_{\gamma(IIR)}(k) = \sum_{n=0}^{p_m-1} h_{\gamma(IIR)}(n)\exp(-j2\pi nk/p_m) \qquad for\ k = 0,\ldots,p_m - 1. \qquad (5.19)$$

Let the DFT of a weighting FIR filter impulse response be $H_{\gamma(FIR)}(k)$. As the relationship between $H_{\gamma(IIR)}(k)$ and $H_{\gamma(FIR)}(k)$ is: $H_{\gamma(IIR)}(k) = \dfrac{1}{H_{\gamma(FIR)}(k)}$; the expression for weighted filtering of the DFT coefficients of the residual prototype waveform, $PW_m(k)$, can be written as:

$$PW_m^{(S)}(k) = \frac{PW_m(k)}{H_{\gamma(FIR)}(k)}. \qquad (5.20)$$

Multiply both the numerator and the denominator of the right hand side in Equation (5.20) by the conjugate of $H_{\gamma(FIR)}(k)$:

$$PW_m^{(S)}(k) = \frac{H^*_{\gamma(FIR)}(k) PW_m(k)}{H^*_{\gamma(FIR)}(k) H_{\gamma(FIR)}(k)}. \qquad (5.21)$$

This equation is the basis for perceptually weighted filtering of the residual PW in the DFT domain whereby the distortions due to the impulse response truncation can be avoided.

5.3.2 Codebook Search

The SEW codebook searching procedure in the speech domain is similar to that in the residual domain, however, before calculating the mean squared error, both the extracted residual PWs and the candidate PWs are passed through the weighting synthesis filters to obtain the speech PWs. Thus, the term $PW_m^{(S)}(k)$ is now used instead of the term $PW_m(k)$ in the total mean squared error (Equation (5.16)).

5.4 A-by-S Errored Magnitude Residual Quantisation

In the previous sections, the analysis-by-synthesis quantisations of the prototype waveforms have been represented either in the residual or in the speech domain. The quantisation technique is based on unity approximation of the normalised PW magnitude spectra. In this technique, only information about the SEW is transmitted, while the REW magnitude spectra is based on the SEW behaviours and unity. In a similar way to that discussed in Chapter 4, this section looks at exploiting the five redundant bits in transmitting the gain term to quantise the information of the REW magnitude (the *Error*) in an analysis-by-synthesis prototype waveform quantisation. The *Error* codebook used here is similar to that employed in Chapter 4.

5.4.1 Errored Magnitude Residual Coder

Quantisation of the *Error* can be used for MPW A-by-S coder operating either in the residual domain or in the speech domain. In this section it is used for the residual domain. The architecture of the Errored Magnitude Residual Encoder is similar to that of the Unity Magnitude Residual Encoder shown in Figure 5.3. However, the construction of the REW is not only based on the chosen SEW codebook vector and the Gaussian noise source, but also the *Error* chosen from a 5 bit *Error* codebook. The REW magnitude is calculated using Equation (4.9), the REW phase is derived from Gaussian noise.

The architecture of the Errored Magnitude Residual Decoder is regarded as a part of the Errored Magnitude Residual Encoder. It is similar to that in Figure 5.4, however, the difference here is the use of the *Error* codebook for construction of the REW.

5.4.2 Codebook Search

The *Error* codebook searching algorithm is performed by best matching the *Error* codeword with the ten *Error*s taken from the extracted PWs in each frame. The mean squared error for the codebook search can be regarded as the total mean squared errors. The *Error* quantisation process is to search across the *Error* codebook such that the total mean squared error $E^{(q)}$ calculated from the following equation is minimised:

$$E^{(q)} = \frac{1}{10}\sum_{m=1}^{10}\frac{1}{p_m}\left\{\sum_{k=0}^{p_m-1}\left|[Error(k)+1]-PWmag_m(k)\right|^2\right\} \qquad (5.22)$$

where $Error(k)$ is the *Error* codebook vector and $PWmag_m(k)$ is the magnitude spectra of the m^{th} extracted, normalised PW.

Having the *Error*(k), the REW magnitude spectra can be constructed using Equation (4.9). The REW phase spectra is derived from a Gaussian noise source. The candidate PWs will be built from the chosen SEW and these REWs. The next step is to search the SEW codebook. The codebook search procedure is to find a SEW such that the candidate PWs best match the extracted PWs. This step is the same as that described in Section 5.2.2.

The bit allocation for the coder is shown in Table 5.2. The total number of bits is 60 bits per 25ms frame, and the total rate is 2.4kb/s.

Parameter	Number of bits	Update rate
LSF	30	40 Hz
pitch	7	40 Hz
gain	5	80 Hz
SEW	8	40 Hz
Error	5	40 Hz

Table 5.2 Bit Allocation for the Errored Magnitude Residual Coder

5.5 A-by-S Errored Magnitude Speech Quantisation

This quantisation scheme is a combination of the Unity Magnitude Speech Quantisation and the Errored Magnitude Residual Quantisation wherein the *Error* quantisation is utilised. Thus, this section investigates the use of a weighting synthesis filter and the quantisation of information regarding the error of the unity magnitude approximation in the MPW analysis-by-synthesis quantisation to improve overall perceptual quality. It is possible to use the perceptual weighting in both the SEW codebook search and the *Error* codebook search, however, this would be much more complex. In this

quantisation scheme, the weighting synthesis filter is thus used only for the SEW codebook search. Before searching the SEW codebook, the residual PWs and the extracted PWs are passed through a DFT domain weighting synthesis filter to obtain the speech PWs (as described in Section 5.3.1). The *Error* codebook search is the same as that discussed in Section 5.4.2.

The architecture of the Errored Magnitude Speech Encoder is a combination of the Errored Magnitude Residual Encoder and the Unity Magnitude Speech Encoder. The architecture of the Errored Magnitude Speech Decoder is the same as that of the Errored Magnitude Residual decoder. Compared to the three previous schemes, this quantisation scheme has the drawback of higher complexity. The bit allocation for this scheme is the same as that of the Errored Magnitude Residual Quantisation scheme and is given in Table 5.2.

5.6 Experimental Results

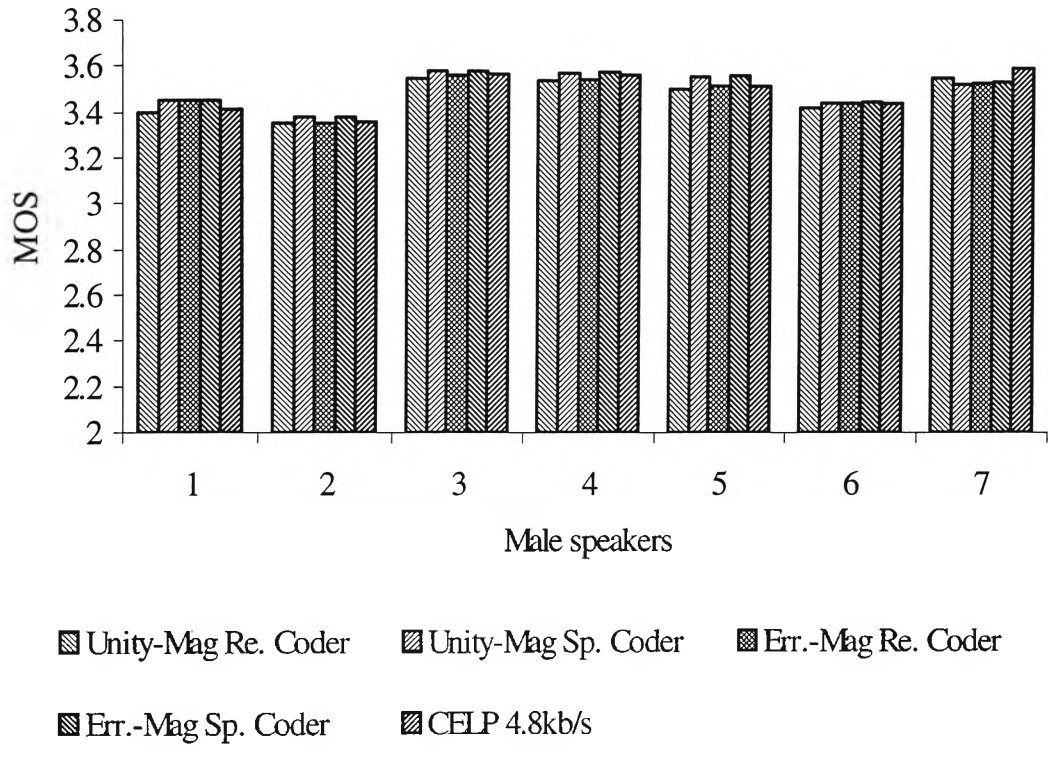
The four MPW A-by-S coders were tested using the MOS criterion. Fourteen TIMIT speech files consisting of seven male and seven female speakers, which were presented in Chapter 4, were used for these tests. The listeners were required to listen and rank the speech test files on the MOS standard table. These tests included the US 1016 Federal Standard CELP-4.8kb/s as a reference.

The results are presented in Table 5.3 and shown as bar charts in Figure 5.8. From the table and the bar charts, it can be seen that on 14 reference speech files, the MOS for the MPW A-by-S coders are close to those for the US 1016 Federal Standard CELP-4.8kb/s. The average MOS for the Unity Magnitude Residual Coder, Unity Magnitude Speech Coder, Errored Magnitude Residual Coder, Errored Magnitude Speech Coder, and the US 1016 Federal Standard CELP-4.8kb/s coder are: 3.501, 3.550, 3.520, 3.556, and 3.536 respectively.

The overall MOS for the four MPW A-by-S coders are equivalent to that for the CELP-4.8kb/s; and is within 0.1 higher than that for the MPW Open Loop coders discussed in Chapter 4. These results show that the prototype waveform coding technique using analysis-by-synthesis architecture can provide a perceptual quality improvement over that using open loop architecture.

Testing speech files	Unity-Mag Re. Coder	Unity-Mag Sp. Coder	Err.-Mag Re. Coder	Err.-Mag Sp. Coder	CELP 4.8kb/s
male 1	3.400	3.448	3.450	3.45	3.410
male 2	3.350	3.377	3.350	3.377	3.360
male 3	3.550	3.580	3.560	3.580	3.570
male 4	3.540	3.575	3.550	3.580	3.570
male 5	3.510	3.560	3.520	3.570	3.520
male 6	3.430	3.450	3.450	3.460	3.450
male 7	3.560	3.533	3.540	3.550	3.610
female 1	3.500	3.520	3.500	3.520	3.530
female 2	3.450	3.480	3.450	3.500	3.420
female 3	3.550	3.680	3.580	3.680	3.650
female 4	3.540	3.600	3.610	3.600	3.590
female 5	3.570	3.643	3.600	3.650	3.630
female 6	3.570	3.680	3.590	3.680	3.670
female 7	3.500	3.575	3.530	3.580	3.530
average results	3.501	3.550	3.520	3.556	3.536

Table 5.3 MOS Results of the MPW A-by-S Coders Compared with the CELP-4.8kb/s Coder (US 1016 Federal Standard)



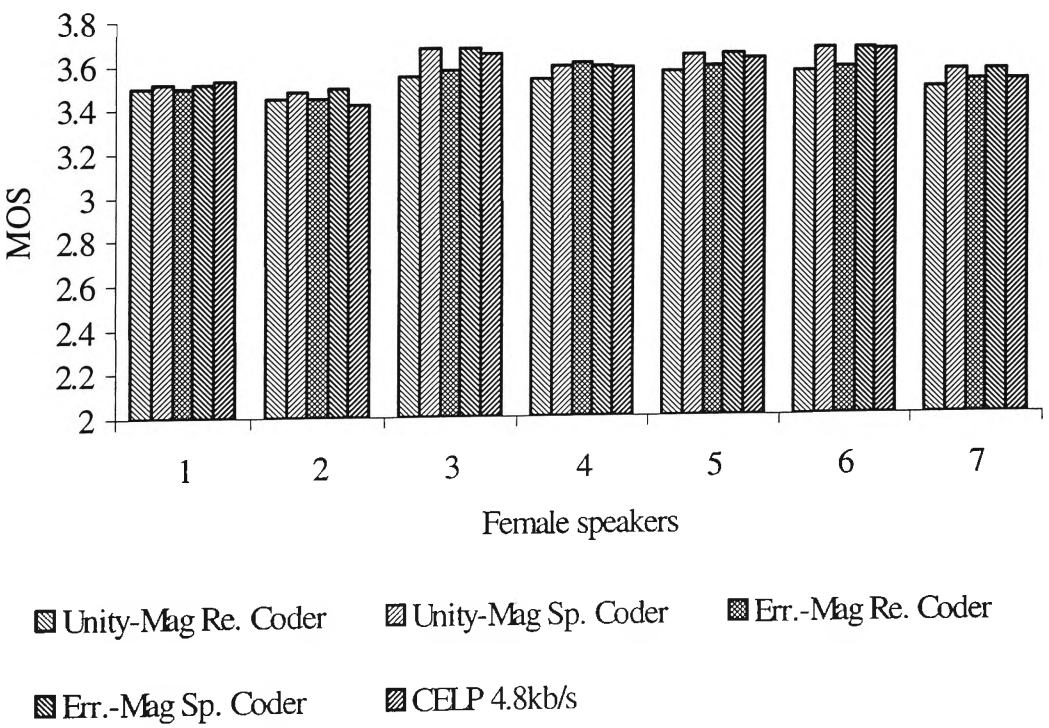


Figure 5.8 Bar Charts of the MOS Test Results of the MPW A-by-S Coders Compared with the CELP-4.8kb/s Coder (US 1016 Federal Standard)

In most cases, the listeners preferred the smoother, clearer speech coded by the MPW A-by-S coders to the speech coded by the MPW open loop coders. Amongst the MPW A-by-S coders, the MOS for the Unity Magnitude Residual Coder was 0.05 less than that for the Unity Magnitude Speech Coder, and 0.02 less than that for the Errored Magnitude Residual Coder. The MOS for the Errored Magnitude Speech Coder was equivalent to that for the Unity Magnitude Speech Coder. These results indicated that the use of perceptual weighting in the SEW codebook search made certain improvements to overall perceptual quality.

5.7 Conclusion

This chapter has described the analysis-by-synthesis quantisation technique for prototype waveform. Based on this, four MPW analysis-by-synthesis coders were developed, which exploited either the perceptual weighting or the use of redundant bits in transmission of the gain term for transmitting information regarding the error in the unity magnitude approximation.

For the SEW codebook search to be more effective, the perceptual coding was utilised whereby the candidate PWs and the extracted PWs were passed through a perceptually weighted filter to obtain speech PWs before the calculation of the mean squared error. This was investigated in the Unity Magnitude Speech Coder. The quantisation of the difference between the actual magnitude of the normalised PW and unity was investigated in the Errored Magnitude Residual Coder; while the Errored Magnitude Speech Coder is a combination of the two above coders. The MOS results show that analysis-by-synthesis quantisation provides certain speech quality improvements over open loop quantisation; and the use of perceptual weighting in the SEW codebook search proved a good method for speech quality enhancement.

Further, the analysis-by-synthesis quantisation technique has shown important advantages over the open loop quantisation technique, in particular:

- Rather than matching the SEW codebook vector and the extracted SEW, the quantisation procedure chooses the SEW codebook vector on the basis of best matching the extracted PWs and the PWs, which it would produce. Thus, the quantisation performance and the speech quality is improved over open loop quantisation.

- The coding technique can be incorporated with a perceptual weighted filter to enhance the overall perceptual quality of the coded speech.

- The MPW analysis-by-synthesis coders are independent from the definition of the SEW, thus it is possible to exploit different sorts of SEW such as the SEW extracted by a low-pass filter with a cut-off frequency of 20Hz [42], or the SEW calculated as an average PW [41].

Chapter 6: Conclusions and Further Work

This thesis has considered quantisation mechanisms in the Multi-Prototype Waveform coding for speech coding at bit rates as low as 2.4kb/s for telecommunications and digital mobile radio satellite communication systems. Based on the proposed coding algorithms, performance of the MPW coders is equivalent to the US 1016 CELP-4.8kb/s Federal Standard. The coded speech sounds more natural than that coded by the CELP-4.8kb/s coder. This chapter reviews the coding techniques and suggests possible further work arising from this thesis.

6.1 Open Loop Quantisation

The Open Loop Quantisation technique was proposed for coding speech at bit rates of 2.4kb/s. The technique was based on exploiting the periodic property of speech whereby it extracted prototype waveforms (PWs) and interpolated between them. For coding at low bit rates, the PWs were required to be decomposed into different components, which can be quantised separately according to their characteristics. This technique, inherently, used the SEW/REW paradigm for decomposition of the PWs. The SEW is a slowly

evolving component, it dominates during voiced speech. The REW is a rapidly evolving component, it dominates during unvoiced speech. At bit rates as low as 2.4kb/s, it has been found that the quantisation of the SEW using an 8 bit SEW codebook for both SEW magnitude and SEW phase spectra was optimum. However, at higher bit rates it is preferable to design separate codebooks for them. Because of random noise characteristics, the REW phase spectra was not quantised and thus recovered by using Gaussian noise. Nevertheless, the REW magnitude spectra was quantised by either the Unity Magnitude Quantisation scheme or the Errored Magnitude Quantisation scheme. In the former, the REW magnitude was not quantised, but recovered on the assumption that the magnitude of the normalised PW is flat and equal to unity. With an aim of improving overall perceptual speech quality, the latter quantised the difference between the actual magnitude of the normalised PW and unity (named as *Error*); thus the REW magnitude was recovered using such information, unity and SEW magnitude.

Two 2.4kb/s MPW coders were developed using the Open Loop Quantisation technique. The MOS test results show that the coded speech was close to that generated by the CELP-4.8kb/s coder (developed similarly to US 1016 Federal Standard). The coded speech quality was mainly dependent on the SEW, while its naturalness was determined by the REW. To achieve coded natural speech it was necessary to control the contribution of the REW to the whole PW. The design of the Errored Magnitude Quantisation scheme aimed at gaining overall perceptual quality, however, the MOS results show that the coded speech was only slightly improved over that coded using the Unity Magnitude Quantisation scheme.

The SEW codebook search made a significant contribution to the perceptual speech quality due to the fact that the voicing of speech is determined by the SEW. In the open loop quantisation, it was performed by a direct search. For

more effective searching, it is possible to use an analysis-by-synthesis architecture. Chapter 5 considered this new coding structure based on these conclusions.

6.2 Codebook Solution

One of the keys to achieving high quality speech in this work was the codebook solution for the SEW and the REW/*Error*. The variation of the PW length, and therefore the REW/SEW length, was a problem in designing the codebooks. In the case of the SEW codebook, this problem was solved simply by zero harmonic padding to the training vectors such that the length of each training vector is equal to a chosen standard length of 148. Each codebook vector had two sections: the first section, 74 SEW magnitude coefficients and the second, 74 SEW phase coefficients. In this work, the codebook size of 256 vectors (8 bits) containing 40 vectors for SEW of unvoiced speech and 216 vectors for SEW of voiced speech was found to be an optimal solution for coding at bit rates of 2.4kb/s. Unlike the SEW codebook, the codebook for the *Error* contains the magnitude only. Each codebook vector is 74 magnitude coefficients.

6.3 Analysis-by-Synthesis Quantisation

Chapter 5 introduced an Analysis-by-Synthesis architecture for prototype waveform quantisation in MPW coding. In this quantisation technique, the SEW codebook was searched on the basis of matching the extracted PW with the candidate PW constructed from the SEW codebook vector. The PW was not decomposed into two components: a SEW and a REW as in Open Loop Quantisation. Rather than matching the SEW, the SEW codebook search was

based on matching the PWs, the perceptual quality was thus significantly improved over the Open Loop Quantisation.

Based on the analysis-by-synthesis quantisation architecture, four MPW coders were developed. The Unity Magnitude Residual Coder operated in the residual domain using the assumption that the magnitude of the normalised PW is unity. The Unity Magnitude Speech Coder also used that assumption, however, it represented the PW in the speech domain and used a weighting synthesis filter for perceptually enhanced coding. The Errored Magnitude Residual Coder quantised the information regarding Error (*Error*) in the unity approximation as information of the REW magnitude. Similar to this coder, the Errored Magnitude Speech Coder transmitted the *Error* information, however, rather than operating in the residual domain it worked in the speech domain, and utilised the perceptual coding.

The MOS tests show that the Analysis-by-Synthesis Based MPW Coders provide a significant improvement to the perceptual quality of speech over the Open Loop Based MPW Coders. Amongst them, the Unity Magnitude Speech Coder produced better quality speech than that generated by the Unity Magnitude Residual Coder. While the Errored Magnitude Residual Coder produced coded speech with an insignificant improvement over that by the Unity Magnitude Residual Coder. The quality of the speech generated by the Errored Magnitude Speech Coder was shown to be close to that produced by the Unity Magnitude Speech Coder.

Analysis-by-Synthesis Quantisation has certain advantages over Open Loop Quantisation. Firstly, the PW decomposition complexity is avoided. Secondly, the MPW analysis-by-synthesis coders are independent from the definition of the SEW, thus, various codebooks of SEW can be used. For coding at higher bit rates it is convenient for transmitting the SEW at higher update rates, rather than 40Hz in the 2.4kb/s coders. Finally, for the SEW codebook search, the

incoming PW and the candidate PW can be represented either in the residual domain or in the speech domain. A weighting synthesis filter, thus, can be used to improve the quantisation performance.

From these investigations the conclusions drawn are that the MPW Analysis-by-Synthesis Coding is a promising method because of its perceptual speech quality improvement and also its advantages over the MPW Open Loop Coding techniques. The use of perceptual coding in the Analysis-by-Synthesis Based MPW Coders proved a novel method in obtaining further quality improvement.

6.4 Summary

In summary, the motivation for the work described in this thesis came from the current demand for high quality speech coding at low bit rates for telecommunication and digital mobile telephone networks. This thesis has proposed two quantisation techniques for Prototype Waveforms: the MPW Open Loop Quantisation and the MPW Analysis-by-Synthesis Quantisation. Both of the techniques were performed in the DFT domain. For these quantisations to be successful, the codebook solutions for the SEW/REW were described. The Open Loop Quantisation decomposed the PW into two distinct components, SEW/REW, and separately quantised them according to their characteristics. The Analysis-by-Synthesis Quantisation did not decompose the PW. The quantisation algorithm searched the SEW codebook by matching either the residual PWs or the speech PWs. Thus, it allowed the MPW coders to be incorporated with a perceptually weighted synthesis filter for further speech quality enhancements. Because of these advantages, the MPW Analysis-by-Synthesis Quantisation could be considered as a promising and realistic quantisation technique for prototype waveforms.

Areas for further work would be the real-time implementation of the 2.4kb/s Analysis-by-Synthesis Based MPW coder. For real-time implementation, it is essential to reduce the complexity of the algorithm. Research on MPW Analysis-by-Synthesis Quantisation for higher bit rates is also an attractive area. At higher bit rates, it is possible to increase the update rate of the SEW and REW quantisation, therefore, coded speech quality could be improved. At low bit rates it is believed that the use of improved bit allocation schemes could lead to improvements in Multi-Prototype Waveform coding. In addition, the application of sub-band coding in Multi-Prototype Waveform coding is another realistic area for future work.

Chapter 7: References

[1] H. Gondokusumo, "Design and Implementation of M-band Perfect Reconstruction Parallel QMF Banks", ME Thesis, University of Wollongong, Australia, 1990.

[2] M.R. Schroeder, and B.S. Atal, "Code-Excited Linear Prediction (CELP): High-Quality Speech at Low Bit Rates", *Proc. IEEE Int. Conf. on Acoust., Speech and Signal Proces.*, pp. 937-940, 1985.

[3] J.H. Chen, R.V. Cox, Y.C. Lin, N.S. Jayant and M.J. Melchener, "A Low-Delay CELP Coder for the CCITT 16kb/s Speech Coding Standard", *IEEE Journal on Sel. Areas in Comms.*, Vol. 10, No. 5, pp. 830-849, June 1992.

[4] P. Vary, R. Hellwig, R.J. Slyter, C. Galland, and M. Rosso, "Speech Codec for the European Mobile Radio System", *Proc. IEEE Int. Conf. on Acoust., Speech and Signal Proces.*, pp. 227-230, New York, USA, 1988.

[5] European Telecommunications Standards Institute Technical Commitee, "Recommendation 06.10: GSM Full-Rate Speech Transcoding", Version 3.2.0, Jan. 1990.

[6] I. A. Atkinson, A. M. Kondoz, B. G. Evans, "Time Envelope Vocoder, a New LP Based Coding Strategy for Use at Bit Rates of 2.4 kb/s and Below", *IEEE Journal on Sel. Areas in Comms.*, Vol. 13, No. 2, pp. 449-457, Feb. 1995.

[7] U.S. National Communication System, Washington, D.C., "Proposed Federal Standard 1016, Second Draft", Nov. 1989.

[8] U.S. National Communications System Washington, D.C., "Details to Assist in Implementation of Federal Standard 1016 CELP", Jan. 1992.

[9] T.E. Tremain, "The Government Standard Linear Predictive Coding Algorithm: LPC-10", *Speech Technology*, Vol. 1, No. 2, pp. 40-49, April 1982.

[10] J.M. Tribolet and R.E. Crochiere, "Frequency Domain Coding of Speech", *IEEE Trans. on Acoust., Speech and Signal Proces.*, Vol. 27, No. 5, pp. 512-530, Oct. 1979.

[11] R. Zelinski and P. Noll, "Adaptive Transform Coding of Speech Signals", *IEEE Trans. on Acoust., Speech and Signal Proces.*, Vol. 25, No. 4, pp. 299-309, August 1977.

[12] N.G. Kingsbury, "Robust 8000bit/s Sub-band Speech Coder", *IEE Proceedings*, Vol. 134, Pt. F, No. 4, pp. 352-366, July 1987.

[13] R. Zemouri, "Design of a Sub-Band Coder For low-Bit Rate Using Fixed and Variable Band Coding Schemes", *Third Annual Int. Conf. on Universal Personal Communications*, pp. 193-198, San Diego, USA, 1994.

[14] M. Brandstein, J. Hardwick and J. Lim, "The Multi-Band Excitation Speech Coders", *Advances in Speech Coding*, Kluwer Academic Publishers, pp. 215-224, 1991.

[15] R.J. McAulay and T. Champion, "Improve Interoperable 2.4kb/s LPC Using Sinusoidal Transform Coder Techniques," *IEEE Int. Conf. on Acoust., Speech and Signal Proces.*, pp. 641-643, USA, April 1990.

[16] R.J. McAulay and T.F. Quatieri, "The Sinusoidal Transform Coder at 2400b/s", *Communications-Fusing Command, Control and Intelligence*, pp. 378-380, Vol. 1, San Diego, USA, Oct. 1992.

[17] R.J. McAulay and T.F. Quatieri, "The Application of Subband Coding to Improve Quality and Robustness of The Sinusoidal Transform Coder", *IEEE Int. Conf. on Acoust., Speech and Signal Proces.*, pp. 439-442, Minneaplis, USA, April 1993.

[18] Y. Shoham, "High-Quality Speech Coding at 2.4 to 4 kbps based on Time-Frequency Interpolation", *Proc. IEEE Int. Conf. on Acoust., Speech and Signal Proces.*, pp. 167-170, 1993.

[19] K. Ozawa, T. Miyano, "4kb/s Improved CELP Coder With Voector Quantisation", *IEEE Int. Conf. on Acoust., Speech and Signal Proces.*, Vol. 1, pp. 589-592, USA, 1991.

[20] J. Haagen, H. Nielsen, S.D. Hansen, "A 2.4kbps High Quality Speech Coder", *IEEE Int. Conf. on Acoust., Speech and Signal Proces.*, Vol. 1, pp. 213-216, USA, 1991.

[21] T. Miyano, M. Serizawa, J. Takizawa, S. Ikeda, and K. Ozawa, "Improved 4.8kb/s CELP Coding Using Two-Stage Vector Quantisation with Multiple Candidates (LCELP)", *IEEE Int. Conf. on Acoust., Speech and Signal Proces.*, pp. 321-324, San Francisco, USA, March 1992.

[22] W. Granzow, B.S. Atal, K.K. Paliwal, and J. Schroeter, "Speech Coding at 4kb/s and Lower Using Single-Pulse and Stochastic Models of LPC Excitation", *IEEE Int. Conf. on Acoust., Speech and Signal Proces.*, Vol. 1, pp. 217-220, Toronto, Canada, May 1991.

[23] M.A. Kohler, L.M. Supplee, and T.E. Tremain, "Progress Towards a New Government Standard 2400 bps Voice Coder", *Proc. IEEE Int. Conf. on Acoust., Speech and Signal Proces.*, pp. 488-491, Detroit, 1995.

[24] W.B. Kleijn, D.J. Krasinski, R.H. Ketchum, "Improved speech quality and efficient vector quantisation in SELP", *Proc. IEEE Int. Conf. on Acoust., Speech and Signal Proces.*, pp. 155-158, 1988.

[25] L.R. Rabiner and R.W. Schafer, "Digital Processing of Speech Signals", *Prentice Hall Signal Processing Series*, 1978.

[26] J.L. Roux and C. Gueguen, "A Fixed Point Computation of Partial Correlation Coefficients", *IEEE Trans. on Acoust., Speech and Signal Proces.*, pp. 257-259, June 1977.

[27] P. Kabal and R.P. Ramachandran, "The Computation of Line Spectral Frequencies Using Chebyshev Polynomial", *IEEE Trans. on Acoust., Speech and Signal Proces.*, Vol. 34, No. 6, pp. 1419-1426, Dec. 1986.

[28] J.P. Campbell, V.C. Welch, and T. Tremain, "An expandable error-protected 4800 bps CELP coder (U. S. Federal Standard 4800 bps voice coder)", *Proc. IEEE Int. Conf. on Acoust., Speech and Signal Proces.*, pp. 735-738, Glasgow, Scotland, May 1989.

[29] J.P. Campbell, V.C. Welch, and T. Tremain, "The DOD 4.8kbps standard (Proposed federal standard 1016)", *Advances in Speech Coding*, (B.S. Atal, V. Cuperman, and A. Gersh, eds.), pp. 121-133, Dordrecht, Holland: Kluwer Academic Publishers, 1991.

[30] F.J. Casajus-Quiros, L.A. Hernandez-Gomez, and C. Garcia-Mateo, "Analysis and Quantisation Procedure for a Real-Time Implementation of a 4.8kb/s CELP coder", *IEEE Int. Conf. on Acoust., Speech and Signal Proces.*, pp. 261-264, Albuquerque, USA, April 1990.

[31] I.S. Burnett, "Hydbrid Techniques for Speech Coding", PhD Thesis, University of Bath, England, 1992.

[32] D.J. Rahikka, T.E. Tremain, V.C. Welch, and J.P. Campbell, "CELP Coding For Land Mobile Radio Applications", *IEEE Int. Conf. on Acoust., Speech and Signal Proces.*, pp. 465-468, Albuquerque, USA, 1990.

[33] G. Yang, H. Leich, and R. Boite, "Voiced speech coding at very low bit rates based on forward-backward waveform prediction (FBWP)", *IEEE Trans. on Speech and Audio Proces.*, Vol. 3, No. 1, pp. 40-47, 1995.

[34] W.B. Kleijn "Continuous Representations in Linear Predictive Coding", *Proc. IEEE Int. Conf. on Acoust., Speech and Signal Proces.*, pp. 201-204, 1991.

[35] W.B. Kleijn, "Speech Coding Below 4kb/s Using Waveform Interpolation", *IEEE Personal Communication Services*, pp. 1879-883, Phoenix, USA, Dec. 1991.

[36] W.B. Kleijn, "Encoding Speech Using Prototype Waveforms", *IEEE Trans. on Speech and Audio Proces.*, Vol. 1, pp. 386-399, 1993.

[37] I.S. Burnett, R.J. Holbeche, "A Mixed Prototype Waveform/CELP Coder for sub 3kb/s", *Proc. IEEE Int. Conf. on Acoust., Speech and Signal Proces.*, pp. 175-178, Minneapolis, 1993.

[38] K. Tang and B. Cheetham, "Variable Frame Length Prototype Waveform Interpolation for Low Bit Rate Speech Coding", in *IEEE Colloquium 1993/234*, 1993.

[39] Y. Tanaka and H. Kimura, "Low Bit Rate Speech Coding Using a Two-Dimensional Transform of Residual Signals and Waveform Interpolation", *Proc. IEEE Int. Conf. on Acoust., Speech and Signal Proces.*, pp. 173-176, 1994.

[40] W.B. Kleijn, and J. Haagen, "Waveform Interpolation for Coding and Synthesis", in *Speech Coding and Synthesis*, ed. W.B. Kleijn and K.K. Paliwal, Elsewier, 1995.

[41] I.S. Burnett, G.J. Bradley, "New techniques for Multi-Prototype Waveform Coding at 2.48kb/s", *Proc. IEEE Int. Conf. on Acoust., Speech and Signal Proces.*, pp. 261-264, Detroit, 1995.

[42] W.B. Kleijn, J. Haagen, "Transformation and Decomposition of Speech Signals for Coding", *IEEE Signal Proces. Letters*, Vol. 1, No. 9, pp. 136-138, 1994.

[43] W.B. Kleijn and J. Haagen, "A Speech Coder on Decomposition of Characteristic Waveforms", *Proc. IEEE Int. Conf. on Acoust., Speech and Signal Proces.*, pp. 508-511, Detroit, 1995.

[44] I.S. Burnett, G.J. Bradley, "Low Complexity Decomposition and Coding of Prototype Waveforms", *Proc. IEEE Workshop on Speech Coding for Telecommunications*, pp. 23-24, Annapolis, 1995.

[45] W.B. Kleijn, Y. Shoham, D. Sen, and R. Hagen, "A Low-Complexity Waveform Interpolation Coder", *Proc. IEEE Int. Conf. on Acoust., Speech and Signal Proces.*, Vol. 1, pp. 212-215, Atlanta, USA, 1996.

[46] P.C. Chang, R.M. Gray, and J. May, "Fourier Transform Vector Quantisation for Speech Coding", *IEEE Trans. on Communications*, Vol. 35, No. 10, pp. 1059-1068, Oct. 1987.

[47] A. Gersho and R.M. Gray, "Vector Quantisation and Signal Compression", Kluwer Academic, 1992.

[48] C.P. Smith, "Voiced Communications Method Using Pattern Matching for Data Compression", *Journal Acoust. America,* pp. 850(A), 1963.

[49] A. Buzo, A.H. Gray, Jr., R.M. Gray, and J. D. Markel, "Speech Coding Based upon Vector Quantisation", *IEEE Trans. on Acoust., Speech and Signal Proces.*, pp. 562-574, Oct. 1980.

[50] P.C. Cosman, "Perceptual Aspects of Vector Quantisation", PhD dissertation, Stanford Univ., Standford, CA.

[51] Y. Linde, A. Buzo and R.M. Gray, "An Algorithm for Vector Quantizer Design", *IEEE Trans. on Communications*, pp. 84-95, January 1980.

[52] J. Makhoul, S. Roucous, H. Gish, "Vector Quantisation in Speech Coding", *Proceeding of the IEEE*, pp. 1551-1588, Nov. 1985.

[53] R.M. Gray, "Vector Quantisation", *IEEE ASSP Magazine*, pp 4-29, April 1984.

[54] T.D. Lookabaugh, R.M. Gray, "High Resolution Quantisation Theory and the Vector Quantiser Advantage", *IEEE Trans. on Inform. Theory*, Vol. 35, No. 5, pp. 1020-1033, Sept. 1989.

[55] J.K. Flanagan, D.R. Morrell, R.L. Frost, C.J. Read, and B.E. Nelson, "Vector Quantisation Codebook Generation Using Simulated

Annealing", *Proc. IEEE Int. Conf. on Acoust., Speech and Signal Proces.*, pp. 1759-1762, Glasgow, Scotland, May 1989.

[56] D.P. Connors and P.R. Kumar, "Simulated Annealing and Balance of Recurrence Order in Time-Inhomogeneous Markov Chains", *Proc. of the 26th Conference on Decision and Control*, pp. 2261-2263, Dec. 1987.

[57] S.R. Quackenbush, T.P. Barnwell III, and M.A. Clements, "Objective Measures of Speech Quality", *Prentice-Hall Signal Processing Series*, 1988.

[58] N.S. Jayant and P. Noll, "Digital Coding of Waveforms: Principles and Applications to Speech and Video", *Prentice-Hall Signal Processing Series*, 1984.

[59] B.S. Atal and M.S. Schroeder, "Predictive Coding of Speech Signals and Subjective Error Criteria", *IEEE Trans. on Acoust., Speech and Signal Proces.*, Vol 27, No. 3, pp. 247-254, June 1979.

[60] Y. Shoham, B.S. Atal, V. Cuperman, and A. Gersho, eds "Constrained-Excitation Coding of Speech at 4.8kb/s", *Advances in Speech Coding*, pp.339-348, Holland, 1991.

[61] N. Kitawaki, M. Honda and K. Itoh, "Speech Quality Assessment for Speech Coding Systems", *IEEE Communications Magazine*, Vol. 22, No. 10, pp. 26-33, Oct. 1984.

[62] N. Kitawaki, K. Itoh, M. Honda, K. Kakehi, "Comparision of Objective Speech Quality Measures for Voiceband Codecs", *Proc. IEEE Int. Conf. on Acoust., Speech and Signal Proces.*, pp. 1000-1003, 1982.

[63] N. Kitawaki, H. Nagabuchi and K. Itoh, "Objective Quality Evaluation for Low-Bit-Rate Speech Coding Systems", *IEEE Journal Sel. Areas in Comms.*, Vol. 6, No. 2, pp. 242-247, Feb. 1988.

[64] A.H. Gray, Jr. and J.D. Markel, "Distance Measures for Speech Processing", *IEEE Trans. on Acoust., Speech and Signal Proces.*, Vol. 24, No. 5, pp. 308-391, Oct. 1976.

[65] I.L. Panzer and A.D. Sharpley, "Comparison of Subjective Testing Methodologies for Speech Quality Evaluation", *Proc. IEEE Workshop on Speech Coding for Telecommunications: Digital Voice for the Nineties*, pp. 93-95, Whistler, B.C., Canada, Sept. 1991.

[66] J.G. Proakis and D.G. Manolakis, "Digital Signal Processing, Principles, Algorithms and Applications", *Macmillan Signal Processing Series*, 1992.

[67] R.P. Ramachandran, M.M. Sondhi, N. Seshadri, B.S. Atal, "A two Codebook Format for Robust Quantisation of Line Spectral Frequencies", *IEEE Trans. on Speech and Audio Proces.*, Vol. 3, No. 3, pp. 157-167, May 1995.

[68] S. Kadambe, and G.F. Boudreaux-Bartels, "Application of Wavelet Transform for Pitch Detection of Speech Signals", *IEEE Trans. on Inform. Theory*, Vol. 38, No. 2, pp. 917-924, March 1992.

[69] M.E. Hernanderz-Diaz Huici, and J.V. Lorenzo Ginori, "Combined Algorithm for Pitch Detection of Speech Signals", *Electronics Letters*, Vol. 31, Iss. 1, pp. 15-16, January 1995.

[70] I.A. Atkinson, A.M. Kondoz, and B.G. Evans, "Pitch Detection of Speech Signals Using Segmented Autocorrelation", *Electronics Letters*, Vol. 31, Iss. 7, pp. 533-535, March 1995.

[71] J.J. Dubnowski, R.W. Schafer and L.R. Rabiner, "Real Time Hardware Pitch Detector", *IEEE Trans. on Acoust., Speech and Signal Proces.*, Vol. 24, No. 1, pp. 2-9, Feb. 1976.

[72] P. Gambino, and I.S. Burnett, "Pitch Detection based on Prototype Waveforms", *IEEE Int. Symposium on Signal Proces. and its Application*, Gold Coast, Australia, 1996.

[73] I.S. Burnett, and P. Gambino, "Low-Delay Pitch Detection using Dynamic-Programming/Viterbi Techniques", *IEEE Int. Symposium on Signal Proces. and its Application*, Gold Coast, Australia, 1996.

[74] D.H. Pham and I.S. Burnett, "Quantisation Techniques for Prototype Waveforms", *IEEE Int. Symposium on Signal Proces. and its Applications*, Gold Coast, Australia, 1996.

[75] J.H. Chen, and A. Gersho, "Real Time Vector APC Speech Coding at 4800b/s With Adaptive Postfiltering", *IEEE Int. Conf. on Acoust., Speech and Signal Proces.*, pp. 2185-2188, Dallas, USA , April 1987.

[76] J.H. Chen and A. Gersho, "Adaptive Postfiltering for Quality Enhancement of Coded Speech", *IEEE Trans. on Speech and Audio Proces.*, Vol. 3, No. 1, pp. 59-71, Jan. 1995.

[77] P. Kabal, F.M. Wang, D. O'Shaughnessy and R.P. Ramachandran, "Adaptive Postfiltering for Enhancement of Noisy Speech in The Frequency Domain", *IEEE Int. Symposium on Circuit and Systems*, pp. 312-315, Singapore, June 1991.

[78] Y. Jiang and V. Cuperman, "Encoding Prototype Waveforms Using a Phase Codebook", *Proc. IEEE Speech Coding Workshop*, pp. 79-80, Annapolis, MD, 1995.

[79] G. Kubin, B.S. Atal, and W.B. Kleijn, "Performance of Noise Excitation for Unvoiced Speech", *Proc. IEEE Workshop on Speech Coding for Telecommunications*, Sainte-Adele, pp. 35-36, 1993.

[80] K.K. Paliwal, and B.S. Atal, "Efficient Vector Quantisation of LPC Parameters at 24 Bits/Frame", *IEEE Trans. on Speech and Audio Proces.*, Vol. 1, No. 1, pp. 3-14, Jan. 1993.

[81] P. Kroon and E.F. Deprettere, "A class of Analysis-by-Synthesis Predictive Coders for High Quality Speech Coding at Rates Between 4.8 and 16 kbits/s," *IEEE Journal on Sel. Areas in Comms.*, Vol. 6, No. 2, pp. 334-363, Feb. 1988.